

SISTEM PENDETEKSI JENIS BUAH MANGGA MENGUNAKAN METODE *TEMPLATE MATCHING*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Agung Bachtiar S
NIM: 145150300111069



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

SISTEM PENDETEKSI JENIS BUAH MANGGA MENGGUNAKAN METODE *TEMPLATE MATCHING*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Agung Bachtiar S.
NIM : 145150300111069

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Agustus 2018
Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dahnial Syauqy, S.T., M.T., M.Sc.
NIK: 201607 870423 1 002

Dosen Pembimbing II

Tibyani, S.T., M.T
NIP: 19691101 199512 1 002

Mengetahui
Ketua Jurusan Teknik Informatika



Eti Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Juli 2018



Agung Bachtiar S.

NIM: 145150300111069

KATA PENGANTAR

Puja dan puji syukur penulis panjatkan kepada Allah SWT karena berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan penelitian dan Laporan Skripsi untuk memperoleh gelar Sarjana Komputer yang berjudul Sistem Pendeteksi Jenis Buah Mangga Menggunakan Metode *Template Matching*.

Dalam pelaksanaan dan penyusunan laporan skripsi ini, tidak sedikit hambatan yang penulis hadapi. Namun penulis menyadari bahwa kelancaran dalam penyusunan laporan ini tidak lain berkat bantuan, dorongan dan bimbingan dari berbagai pihak, sehingga kendala-kendala yang penulis hadapi dapat teratasi. Penghargaan dan terima kasih yang sebesar-besarnya penulis sampaikan kepada:

1. Kedua orangtua dan saudara kandung penulis yang tiada henti memberikan semangat dan dorongan baik berupa do'a maupun materi selama penulis melakukan penelitian.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika periode 2016 – saat ini.
3. Bapak Sabriansyah Rizqika Akbar, S.T., M.Eng. selaku Ketua Program Studi Teknik Komputer periode 2016 – saat ini.
4. Bapak Dahnial Syauqy, S.T., M.T., M.Sc. selaku pembimbing I yang telah banyak membantu memberikan arahan dan bimbingan selama pengerjaan skripsi.
5. Bapak Tibyani, S.T, M.T selaku pembimbing II yang telah banyak membantu memberikan arahan dan bimbingan selama pengerjaan skripsi.
6. Nur Ananda C yang selalu memberi suntikan motivasi kepada penulis untuk segera menyelesaikan pengerjaan skripsi ini.
7. Teman-teman Teknik Komputer 2014 yang selalu memotivasi penulis untuk segera menyelesaikan penulisan skripsi ini.
8. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah berperan dalam penyelesaian skripsi ini.

Akhir kata penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan yang perlu disempurnakan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan. Semoga skripsi ini bermanfaat bagi semua pihak yang membacanya.

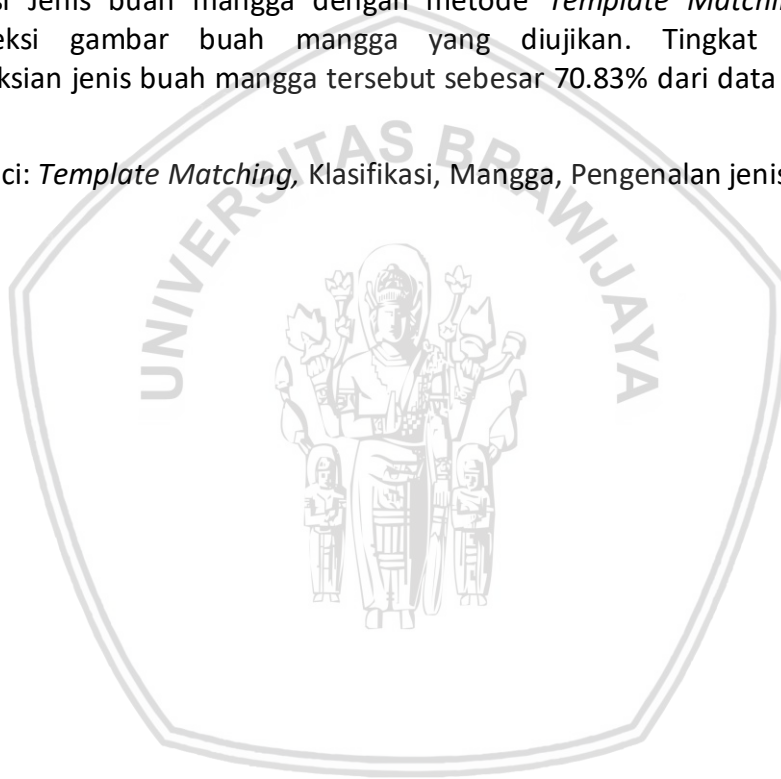
Malang, 25 Juli 2018
Penulis

agungbachtiar07@gmail.com

ABSTRAK

Algoritma *Template Matching* merupakan metode sederhana yang digunakan untuk mengenali pola pada sebuah citra. Metode *Template Matching* sering digunakan untuk mengidentifikasi citra karakter huruf, angka, sidik jari dan aplikasi-aplikasi pencocokan citra lainnya. Pada Penelitian ini dibuat sebuah sistem untuk mendeteksi jenis buah mangga. Terdapat 3 jenis buah mangga yang dapat dideteksi, yaitu Manalagi, Arumanis, dan Golek. Metode yang digunakan pada aplikasi ini adalah *Template Matching*, yaitu teknik statistik yang digunakan untuk mencari dua variabel/matriks atau lebih yang sifatnya kuantitatif. Algoritma ini mencocokkan setiap pixel pada suatu matriks citra digital dari buah mangga dengan citra yang menjadi template acuan. Hasil dari uji coba menunjukkan bahwa klasifikasi Jenis buah mangga dengan metode *Template Matching* ini dapat mendeteksi gambar buah mangga yang diujikan. Tingkat keberhasilan pendeteksian jenis buah mangga tersebut sebesar 70.83% dari data uji sebanyak 24 citra.

Kata kunci: *Template Matching*, Klasifikasi, Mangga, Pengenalan jenis buah mangga



ABSTRACT

Template Matching Algorithm is a simple method used to recognize patterns in an image. Template Matching Methods are often used to identify letters, numbers, fingerprint and other image matching images. In this research is made a system to detect type of mango fruit. There are 3 types of mangoes that can be detected, namely Manalagi, Arumanis, and Golek. The method used in this application is Template Matching, which is a statistical technique used to find two variables / matrices or more that are quantitative. This algorithm matches each pixel in a digital image matrix of mango from the image that becomes the template reference. The results of the experiments show that the classification of the type of mango fruit with Template Matching method can detect images of tested mangoes. The success rate of mango fruit detection is 70.83% of the test data of 24 images.

Keywords: Template Matching, Classification, Mango, mangoes recognition



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR TABEL.....	4
DAFTAR GAMBAR.....	5
BAB 1 PENDAHULUAN.....	5
1.1 Latar Belakang.....	6
1.2 Rumusan Masalah	7
1.3 Tujuan Penelitian.....	7
1.4 Manfaat Penelitian.....	7
1.5 Batasan Masalah	7
1.6 Sistematika Pembahasan	8
BAB 2 LANDASAN KEPUSTAKAAN	9
2.1 Tinjauan Pustaka	9
2.2 Dasar Teori	12
2.2.1 Mangga Manalagi	12
2.2.2 Mangga Arumanis.....	13
2.2.3 Mangga Golek.....	13
2.2.4 Raspberry Pi 3	14
2.2.5 Webcam Logitech C270	17
2.2.6 OpenCV	18
2.2.7 Template Matching.....	19
BAB 3 METODE PENELITIAN	22
3.1 Studi Literatur.....	23
3.2 Analisis Kebutuhan	23
3.2.1 Kebutuhan Perangkat Keras	23
3.2.2 Kebutuhan Perangkat Lunak.....	24
3.3 Perancangan.....	24
3.4 Implementasi Sistem	24

3.5 Pengujian dan Analisis.....	24
3.6 Kesimpulan	24
BAB 4 REKAYASA KEBUTUHAN	26
4.1 Gambaran Umum Sistem	26
4.2 Analisis Kebutuhan Sistem	26
4.2.1 Kebutuhan Fungsional	26
4.2.2 Kebutuhan Non Fungsional	27
4.2.2. Kebutuhan Perangkat Lunak	28
4.3 Batasan Desain Sistem	28
BAB 5 PERANCANGAN DAN IMPLEMENTASI	29
5.1 Perancangan Sistem	29
5.1.1 Perancangan Purwarupa Alat Pendeteksi Jenis Buah Mangga .	29
5.1.2 Perancangan Perangkat Keras.....	30
5.1.3 Perancangan Perangkat Lunak.....	31
5.2 Implementasi Sistem	35
5.2.1 Implementasi Alat Pendeteksi Jenis Buah Mangga	35
5.2.2 Implementasi Perangkat Keras	36
5.2.3 Implementasi Perangkat Lunak.....	37
BAB 6 PENGUJIAN DAN ANALISIS.....	45
6.1 Pengujian Kamera Logitech C270.....	45
6.1.1 Tujuan Pengujian.....	45
6.1.2 Prosedur Pengujian	45
6.1.3 Hasil dan Analisis Pengujian	46
6.2 Pengujian LED.....	48
6.2.1 Tujuan Pengujian.....	48
6.2.2 Prosedur Pengujian	48
6.2.3 Hasil dan Analisis Pengujian	49
6.3 Pengujian Akurasi Klasifikasi Template Matching.....	50
6.3.1 Tujuan Pengujian.....	50
6.3.2 Prosedur Pengujian	50
6.3.3 Hasil dan Analisis Pengujian Klasifikasi <i>Template Matching</i>	51
6.4 Pengujian Waktu Komputasi Pemrosesan Sistem.....	55
6.4.1 Tujuan Pengujian	55
6.4.2 Prosedur Pengujian	55

6.4.3 Hasil dan Analisis Pengujian	56
BAB 7 PENUTUP	58
7.1 Kesimpulan	58
7.2 Saran	58
DAFTAR PUSTAKA.....	60
LAMPIRAN A SAMPEL BUAH MANGGA.....	62
LAMPIRAN B KODE PROGRAM KESELERUHAN SISTEM PENDETEKSI JENIS BUAH MANGGA MENGGUNAKAN <i>TEMPLATE MATCHING</i>	63



DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	10
Tabel 2.2 Spesifikasi <i>Raspberry Pi 3</i>	16
Tabel 2.3 Spesifikasi webcam Logitech 270	18
Tabel 2.4 <i>Method Template Matching</i> yang disediakan <i>OpenCV</i>	20
Tabel 5.1 Keterangan koneksi <i>pin LED</i> dengan <i>pin GPIO Raspberry</i>	31
Tabel 5.2 Kode Pemrograman Inisialisasi <i>Library</i> Sistem Pendeteksi Jenis Buah Mangga	38
Tabel 5.3 Kode Pemrograman Inisialisasi Pin GPIO	38
Tabel 5.4 Kode Pemrograman Pengecekan <i>Trigger</i> dan Pengambilan gambar .	40
Tabel 5.5 Kode Pemrograman Pembuatan <i>Template</i>	41
Tabel 5.6 Kode Pemrograman <i>Template Matching</i>	43
Tabel 6.1 Hasil pengujian pengambilan gambar dengan kamera Logitech C270 .	46
Tabel 6.2 Hasil pengujian LED	49
Tabel 6.3 Hasil pengujian Klasifikasi <i>Template Matching</i>	51
Tabel 6.4 Kode Program Pengujian Waktu Komputasi	55
Tabel 6.5 Hasil pengujian waktu komputasi	56
Tabel 6.6 Hasil pengujian waktu komputasi berdasarkan jenis.....	57

DAFTAR GAMBAR

Gambar 2.1 Mangga Manalagi.....	12
Gambar 2.2 Mangga Arumanis	13
Gambar 2.3 Mangga Golek	13
Gambar 2.4 Raspberry Pi 3.....	14
Gambar 2.5 <i>Pin GPIO Raspberry Pi 3</i>	14
Gambar 2.6 Skematik Raspberry Pi 3.....	15
Gambar 2.7 <i>Webcam Logitech C270</i>	17
Gambar 2.8 Prinsip Kerja Kamera	17
Gambar 2.9 <i>OpenCV</i>	18
Gambar 3.1 Diagram Alur Metodologi Penelitian	22
Gambar 3.2 Diagram blok Perancangan Sistem.....	24
Gambar 5.2 Diagram Skematik Sistem.....	30
Gambar 5.3 Diagram alir perancangan Pembuatan <i>Template</i>	33
Gambar 5.4 Diagram alir Perancangan Pengenalan Klasifikasi Dengan <i>Template Matching</i>	34
Gambar 5.5 Implementasi Purwarupa Alat Pendeteksi Jenis Buah Mangga.....	36
Gambar 5.6 Implementasi Rangkaian <i>Webcam</i> dan <i>Raspberry Pi</i>	36
Gambar 5.7 Implementasi Rangkaian LED dan Push Button	37
Gambar 5.8 Hasil Gambar <i>RGB</i>	39
Gambar 5.9 Hasil Gambar <i>Grayscale</i>	39
Gambar 5.11 Hasil Beberapa Gambar <i>Template Grayscale</i>	41
Gambar 5.12 Hasil Gambar Deteksi Jenis Mangga Manalagi	42
Gambar 5.13 Hasil Gambar Deteksi Jenis Mangga Arumanis.....	42
Gambar 5.14 Hasil Gambar Deteksi Jenis Mangga Golek.....	43
Gambar 6.1 Hasil Pengujian Waktu Komputasi Pada terminal.....	57

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu negara di Asia Tenggara yang memiliki sumber daya alam yang sangat melimpah, baik itu berupa tanaman, hewani maupun pada bidang pertambangan. Banyak sekali tanaman buah-buahan yang tumbuh subur di tanah Indonesia, seperti salah satunya adalah buah Mangga (*Mangifera Indica*). Mangga merupakan tanaman buah tahunan yang berasal dari negara India, Srilanka dan Pakistan yang kemudian menyebar ke seluruh dunia. termasuk Indonesia (Winda Pratiwi. 2015). Pohon mangga hampir selalu ada di pekarangan-pekarangan rumah masyarakat. Buah mangga dapat tumbuh subur di daerah yang memiliki iklim tropis. Beberapa daerah yang menjadi sentra produksi buah mangga seperti di Probolinggo.

Tumbuhan mangga (*Mangifera indica*) tergolong kelompok buah yang memiliki daging dengan bentuk, ukuran, warna serta citarasa yang beranekaragam. Buah Mangga mengandung banyak vitamin yang sangat dibutuhkan oleh tubuh manusia, antara lain vitamin A, B, C yang tinggi (AAK. 1997). Buah mangga termasuk salah satu jenis buah yang sangat digemari dan sering dikonsumsi oleh masyarakat Indonesia, baik buah yang masih muda sampai yang sudah tua.

Perkembangan ilmu pengetahuan dan teknologi membuat kualitas kehidupan manusia semakin berkembang. Termasuk pada industri pengolahan hasil pertanian dan perkebunan. Dalam pengolahan hasil pertanian dan perkebunan, salah satu tahap yang perlu dilakukan adalah proses pemilihan produk berdasarkan kualitas dan jenisnya.

Berdasarkan observasi yang dilakukan di pasar Blimbing 1 Agustus 2018, para konsumen buah mangga sering dibingungkan dalam membedakan jenis buah mangga. Hal ini disebabkan oleh kemiripan bentuk dan ukuran buah mangga contohnya antara buah manalagi dengan arumanis. Para pengusaha manga juga terkadang masih menggunakan cara tradisional dalam menentukan jenis mangga. Namun upaya tersebut sering tidak akurat dan memiliki penentuan yang berbeda-beda. Perbedaan tersebut diakibatkan karena bedanya persepsi pada setiap orang. Karena perbedaan tersebut terkadang sering terjadi buah manga yang jenisnya tertukar dalam proses *packing*. Hal ini dapat membuat kerugian penjual dan membuat konsumen kecewa karena buah manga yang telah dibeli tidak sesuai dengan keinginan konsumen.

Oleh sebab itu penulis tertarik untuk membangun sebuah alat yang dapat mendeteksi buah mangga berdasarkan jenisnya. Baik mangga jenis Manalagi, Arum Manis, maupun Mangga Golek. Sehingga dengan demikian masyarakat khususnya pengusaha mangga akan mampu mengelompokkan buah mangga berdasarkan jenis yang lebih akurat. Dengan dikelompokkannya buah mangga berdasarkan jenisnya maka selanjutnya akan mempermudah proses pengemasan dan juga dapat menghemat waktu, tenaga dan biaya.

Pembuatan Alat ini memanfaatkan webcam, raspberry pi, dan Open CV sebagai pengolah gambarnya kemudian akan ditampilkan ke layar LCD ataupun menggunakan lampu LED.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, dapat ditarik rumusan masalah sebagai berikut:

1. Bagaimana sistem mendapatkan fitur tampilan fisik buah mangga?
2. Bagaimana sistem dapat mendeteksi jenis- jenis buah mangga?
3. Bagaimana tingkat keakuratan sistem dalam mendeteksi jenis buah mangga?
4. Bagaimanakah performansi Sistem Pendeteksi Jenis Buah Mangga dengan menggunakan metode *Template Matching*?

1.3 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah disampaikan diatas, tujuan penelitian yang ingin dicapai adalah:

1. Untuk mengetahui cara sistem mendapatkan fitur tampilan fisik buah mangga
2. Untuk mengetahui cara sistem dalam mendeteksi jenis – jenis buah mangga
3. Dapat mengetahui ke akurasian sistem dalam mendeteksi jenis buah manga
4. Dapat mengetahui performansi Sistem Pendeteksi Jenis Buah Mangga dengan menggunakan metode *Template Matching*

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah agar semua orang baik petani ataupun konsumen buah mangga mengetahui jenis buah mangga dan mempermudah pedagang buah mangga untuh memilah buah mangga dalam pengepakan.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Terdapat tiga jenis buah mangga, yaitu Manalagi, Arum Manis dan Golek
2. Ukuran maksimal buah mangga yang dapat dideteksi
3. Sistem memerlukan cahaya yang cukup dalam mendeteksi buah mangga

1.6 Sistematika Pembahasan

Sistematika yang digunakan untuk penulisan laporan penelitian ini yaitu:

BAB I PENDAHULUAN

Bab I menjelaskan tentang latar belakang mengapa penelitian ini dibuat, kemudian menentukan rumusan masalah dari latar belakang, tujuan dan manfaat penelitian, batasan penelitian, serta sistematika pembahasan pada laporan yang akan dilakukan oleh peneliti

BAB II LANDASAN KEPUSTAKAAN

Pada bab landasan kepustakaan menjelaskan teori – teori yang berkaitan dengan apa yang akan digunakan oleh peneliti. Selain itu juga membahas penelitian terdahulu yang berkaitan dengan yang dilakukan peneliti.

BAB III METODOLOGI PENELITIAN

Pada bab metodologi penelitian membahas bagaimana alur kerja penelitian yang akan dilakukan peneliti yaitu studi literatur, analisis kebutuhan yang akan digunakan untuk penelitian, dan perancang sistem yang akan dibuat.

BAB IV REKAYASA KEBUTUHAN

Bab rekayasa kebutuhan akan menjelaskan bahan – bahan, perangkat keras ataupun lunak apa saja yang dibutuhkan dalam mengimplementasikan sistem yang akan dibuat, baik dari segi fungsional maupun nonfungsional

BAB V PERANCANGAN DAN IMPLEMENTASI

Bab perancangan dan implementasi menjelaskan tentang proses pembuatan sistem, mulai dari perancangan perangkat keras sampai perancangan perangkat lunak. Pada pengimplementasian sistem, akan dilakukan perancangan sesuai model yang telah ditentukan, dan mampu berjalan sesuai dengan tujuannya atau sesuai dengan apa yang telah diprogram.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis akan ditampilkan hasil dari perancangan sistem yang telah dibuat, dan kemudian akan diuji oleh peneliti untuk di analisis dari segi performa yang diberikan, keakuratan dan kecepatan sistem.

BAB VII PENUTUP

Pada bab penutup ini akan dijelaskan mengenai kesimpulan terhadap hasil dari pengujian yang telah dilakukan oleh peneliti. Selain itu juga terdapat saran yang diberikan oleh penulis untuk pengembangan sistem selanjutnya agar sistem yang dibuat dapat berjalan lebih baik lagi.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Landasan yang penulis gunakan dalam penelitian ini adalah hasil penelitian terdahulu yang telah dilakukan oleh Thomas Oddy Chrisdwianto dengan judul “Perancangan Sistem Deteksi dan Pengenalan Rambu Peringatan Menggunakan Metode *Template Matching*”. Metode yang digunakan pada penelitian tersebut adalah metode *template matching* dengan tingkat keakurasian mencapai 89% (Thomas Oddy Chrisdwianto. 2018).

Kemudian pada penelitian selanjutnya adalah penelitian yang telah dilakukan oleh Rezki Trianto, dkk dengan judul “Klasifikasi Huruf Katakana Dengan Metode *Template Matching Correlation*”. Pada penelitian tersebut juga menggunakan metode *template matching* dengan tingkat keakurasian dari sistem yang dibuat mencapai 85% (Rezki Trianto, dkk).

Landasan selanjutnya adalah “Pengenalan Plat Nomor Kendaraan Menggunakan Metode *Template Matching* Dan Jarak Canberra” yang dilakukan oleh Tito Tri Pamungkas. Sesuai dengan judulnya, Tito menggunakan metode *template matching* dan jarak Canberra. Pada tahap pengujiannya tingkat keakurasian yang dihasilkan dengan menggunakan metode *template matching* sebesar 90%. Sedangkan dengan menggunakan metode Jarak Canberra, diperoleh tingkat keakurasian sebesar 85%. Sehingga pada penelitian tersebut dapat disimpulkan bahwa metode *Template Matching* memiliki tingkat pengenalan yang lebih baik dibandingkan dengan metode jarak Canberra (Tito Tri Pamungkas. 2014).

“Pemanfaatan Metode *Template Matching* untuk Face Tracking secara Real Time di Ruang Tertutup” adalah landasan keempat yang penulis gunakan. penelitian tersebut dilakukan oleh Efraim Anggriyono. Sistem perekaman video otomatis yang dilakukan Efraim diuji dengan 32 subyek dan menghasilkan rata-rata akurasi program (Ap) sebesar 79.95% (Efraim Anggriyono. 2015).

Penelitian yang menjadi landasan berikutnya bagi penulis adalah penelitian dengan judul “Deteksi Osteoporosis Dengan Metode *Template Matching* Pada Citra Sinar Rontgen Tulang Panggul Manusia” yang dilakukan oleh Chairul Umam. Dari pengujian pada citra sinar rontgen tulang panggul yang dilakukan dengan metode alternatif yang dibuat, dapat dimanfaatkan untuk mendeteksi osteoporosis dengan tingkat akurasi sebesar 78.94%. Dimana metode alternatif tersebut adalah perpaduan dari dua metode yaitu metode *Template Matching* dan metode Euclidean Distance (Chairul Umam. 2015).

Penelitian dari Adhitya Wishnu Wardhana dengan judul “Penggunaan Metode Template Matching untuk Identifikasi Kecacatan Pada PCB” adalah landasan yang berikutnya bagi penulis. Pada penelitian tersebut menggunakan dua buah model PCB. Yaitu PCB acuan (master) dan PCB RAM yang cacat (input). Pada hasil penelitian terlihat bahwa metode *template matching* dapat memberikan output berupa identifikasi kecacatan dari PCB input. Sehingga, simulasi proses AOI (Automated Optical Inspection) yang diterapkan dalam aplikasi ini dapat berjalan dengan baik (Adhitya Wishnu Wardhana. 2008).

“Pengenalan Image Wajah Dengan Menggunakan Metode *Template Matching*” yang merupakan hasil penelitian dari Moh. Khayat Subkhan bahan acuan selanjutnya bagi penulis. Dalam penelitian itu menggunakan sampling wajah sebanyak 60 image dan memperoleh tingkat keakurasian sebesar 85%. Faktor yang mempengaruhi hasil pengenalan image wajah pada penelitian ini adalah faktor jarak, pencahayaan, dan pose obyek (Moh. Khayat Subkhan. 2012)

Penelitian Yuni Yamasari judul “Aplikasi Identifikasi Garis Telapak Tangan Berbasis *Template Matching*” merupakan salah satu penelitian yang digunakan sebagai landasan oleh penulis. Pada penelitian ini aplikasi yang dibuat berhasil mengenali garis-garis telapak tangan 7 dari 10 orang. Tingkat kegagalannya di pengaruhi oleh posisi pengambilan citra dan pencahayaan saat registrasi citranya. Sehingga Tingkat keakurasian metode *template matching* pada penelitian ini mencapai 70% (Yuni Yamasari. 2012).

Oleh karena itu pada penelitian ini sistem akan menggunakan metode *template matching*. karena dari penelitian-penelitian sebelumnya diperoleh tingkat keakurasian yang cukup tinggi.

Tabel 2.1 Tinjauan Pustaka

No	Penulis	Judul	Deskripsi	Perbedaan
1	Thomas Oddy Chrisdwianto	Perancangan Sistem Deteksi dan Pengenalan Rambu Peringatan Menggunakan Metode <i>Template Matching</i>	Tingkat keakurasian metode <i>template matching</i> mencapai 89%	Sistem menggunakan matlab sebagai perangkat lunak pengolahan citra
2	Rezki Trianto. dkk	Klasifikasi Huruf Katakana Dengan	Tingkat keakurasian	huruf katakana akan dilakukan

		Metode <i>Template Matching Correlation</i>	metode <i>template matching</i> mencapai 85%	proses operasi "OR" terhadap masing-masing pixelnya sehingga Sakan ditemukan nilai matriks citra baru yang akan dibandingkan dengan data uji yang ada.
3	Tito Tri Pamungkas	Pengenalan Plat Nomor Kendaraan Menggunakan Metode <i>Template Matching</i> Dan Jarak Canberra	Tingkat keakurasian metode <i>template matching</i> mencapai 90%	Menggunakan <i>Template Matching</i> dan jarak canberra sebagai pengenalan
4	Efraim Anggriyono	Pemanfaatan Metode <i>Template Matching</i> untuk <i>Face Tracking</i> secara <i>Real Time</i> di Ruang Tertutup	Tingkat keakurasian metode <i>template matching</i> mencapai 79.95%	Sistem digunakan secara <i>Real Time</i>
5	Chairul Umam	Deteksi <i>Osteoporosis</i> Dengan Metode <i>Template Matching</i> Pada Citra Sinar <i>Rontgen</i> Tulang Panggul Manusia	Tingkat keakurasian metode <i>template matching</i> mencapai 78.94%	Menerapkan metode Euclidean Distance untuk pencocokan deteksi osteoporosis.
6	Adhitya Wishnu Wardhana	Penggunaan Metode <i>Template Matching</i> Untuk Identifikasi Kecacatan Pada <i>PCB</i>	simulasi proses <i>AOI (Automated Optical Inspection)</i> yang diterapkan dalam aplikasi ini dapat berjalan dengan baik	Menggunakan Median Filter untuk melakukan proses penapisan jika citra jika mengandung derau.
7	Moh. Khayat Subkhan	Pengenalan <i>Image</i> Wajah Dengan	Tingkat keakurasian metode <i>template</i>	Jarak eigenvalue dari image testing yang terpendek

		Menggunakan Metode Template Matching	<i>matching</i> mencapai 85%	akan direkomendasikan sebagai image yang paling mendekati faktor kecocokannya
8	Yuni Yamasari	Aplikasi Identifikasi Garis Telapak Tangan Berbasis Template Matching	Tingkat keakurasian metode <i>template matching</i> mencapai 70%	Menggunakan metode Otsu sebagai thresholding untuk membedakan <i>background</i> dan <i>foreground</i>

2.2 Dasar Teori

2.2.1 Mangga Manalagi



Gambar 2.1 Mangga Manalagi

(Sumber: Ir. Pracaya, 2011)

Pada Gambar 2.1 merupakan Gambar Mangga Manalagi yang merupakan buah yang berasal dari Pasuruan. Mangga Manalagi memiliki rasa yang hampir sama dengan mangga golek dan arumanis. Ukuran buah mangga manalagi terbesar diantara mangga unggul lainnya. Beratnya mencapai 900 gram, Panjang buah dapat mencapai 15 cm, lebar 9 cm, dan tebalnya sekitar 8 cm (Sumeru Ashari. 2017). Kulit buah mangga tipis, apabila masak buah akan berwarna kekuning-kuningan, berair, tidak berserat, dan manis. Buah mangga manalagi ini tidak tahan lama dalam penyimpanan. Produksi buah per pohonnya pun lebih sedikit dibanding varietas gadung dan golek. Rendahnya produktivitas mangga manalagi kemungkinan disebabkan karena jumlah bunga *hermaproditnya* sedikit.

2.2.2 Mangga Arumanis



Gambar 2.2 Mangga Arumanis
(Sumber: Ir. Pracaya, 2011)

Pada Gambar 2.2 merupakan gambar Mangga Arumanis yang merupakan buah yang berasal dari Kota Probolinggo. Mangga arumanis memiliki daun kecil, oval, dan tebal, serta tepi daun rata. Berat buah mangga arumanis sekitar 385 - 400 gram. panjang buah 12 -15 cm, dan lebar sekitar 8 cm, dengan tebal 7 cm (Sumeru Ashari. 2017). Daging buahnya mengandung air lebih banyak daging berwarna kuning belerang dan warna kulit pada buah yang masak akan muncul bintik pada pangkal buahnya.

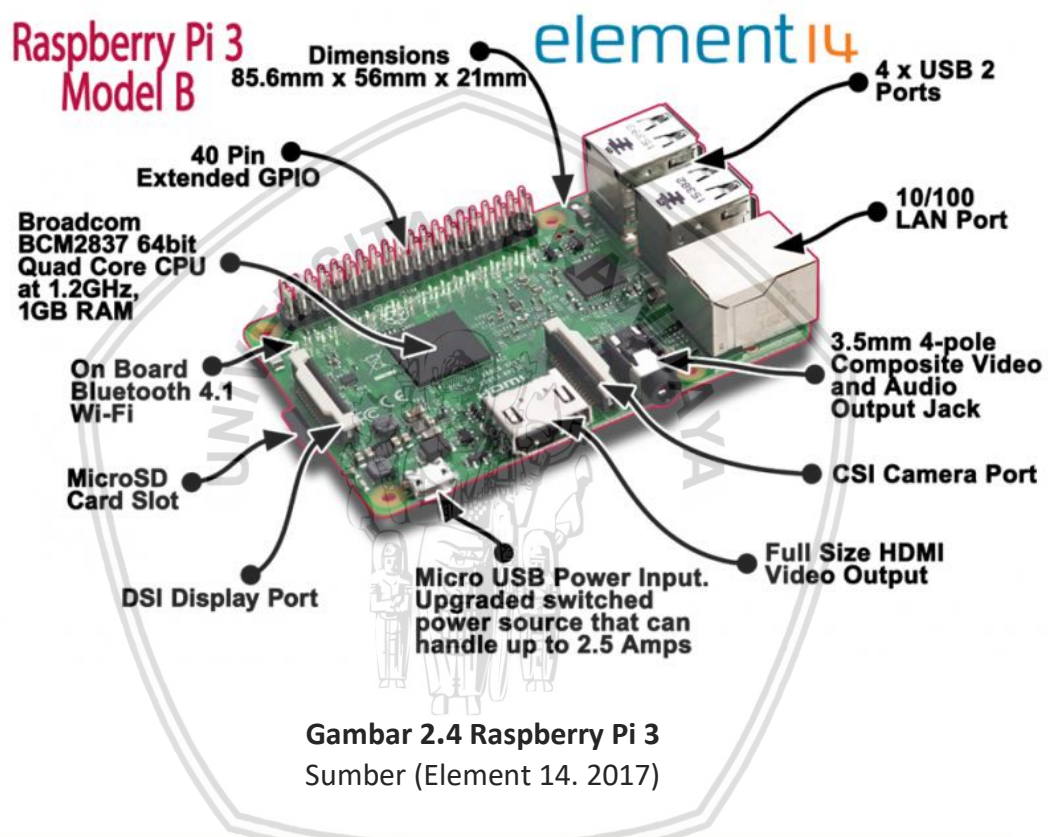
2.2.3 Mangga Golek



Gambar 2.3 Mangga Golek
(Sumber: Ir. Pracaya, 2011)

Mangga golek merupakan buah mangga yang berasal dari seban, Pasuruan. Buah ini memiliki bentuk bulat lonjong agak gepeng dengan berat antara 300 – 700 gram. panjang buah antara 15 – 22 cm, lebarnya 7 – 9 cm dan tebal 6 – 8 cm (Sumeru Ashari. 2017). Buah mangga golek memiliki warna hijau, apabila sudah masak akan berubah kekuning – kuning dan berbintik, kulit buah tebal memiliki daging buah berwarna kuning jernih hingga oranye, manis dan tidak berserat. Apabila mangga golek terlalu masak rasanya menjadi agak kurang enak.

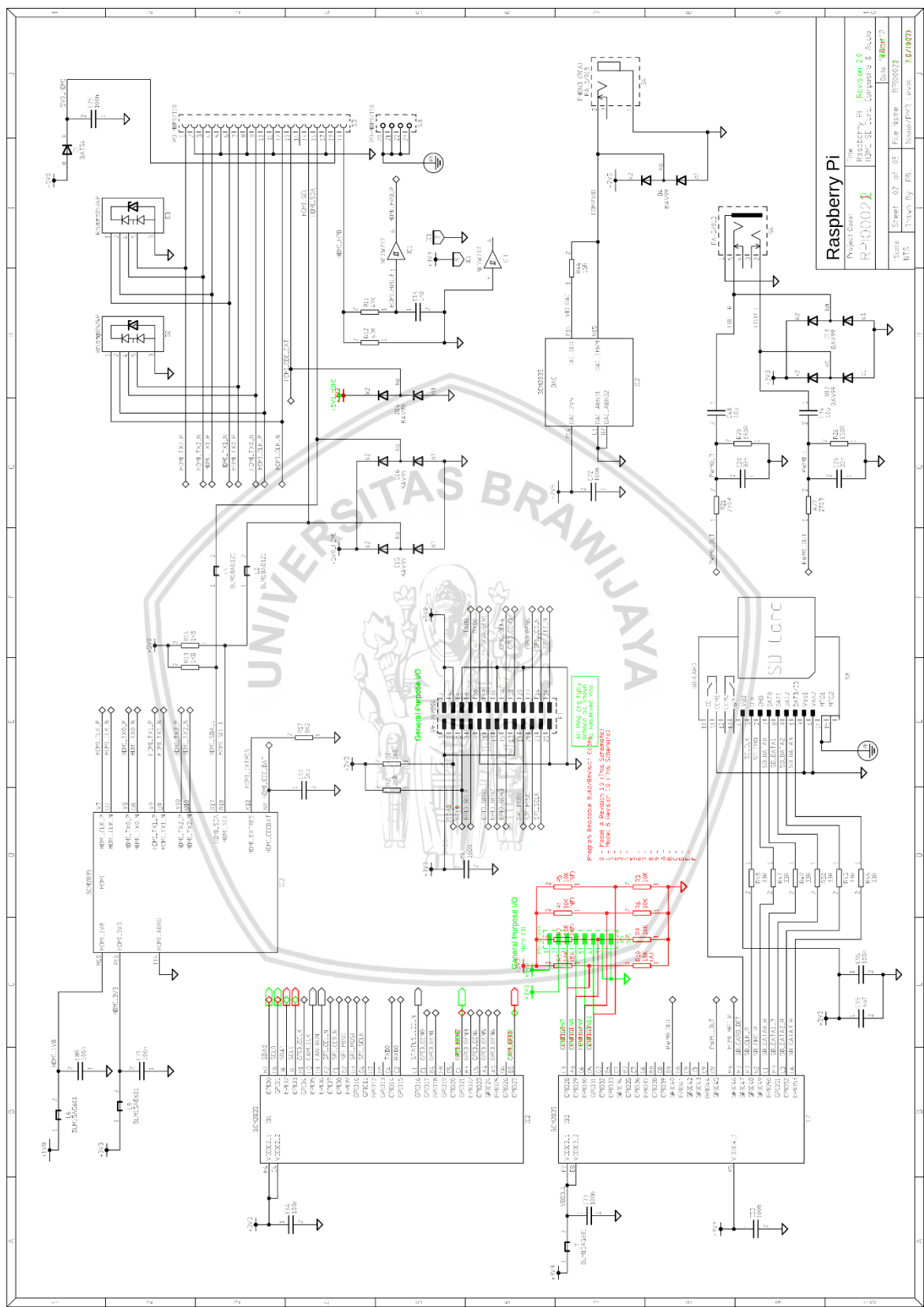
2.2.4 Raspberry Pi 3



Gambar 2.4 Raspberry Pi 3
Sumber (Element 14. 2017)

5V Power	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
5V Power	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
Ground																				
GPIO14 UART0_TXD																				
GPIO15 UART0_RXD																				
GPIO18 PWRM_CLK																				
Ground																				
GPIO23																				
GPIO24																				
Ground																				
GPIO25																				
GPIO8 SPI0_CE0_N																				
GPIO7 SPI0_CE1_N																				
ID_SC DC ID EEPROM																				
Ground																				
GPIO12																				
Ground																				
GPIO16																				
GPIO20																				
GPIO21																				

Gambar 2.5 Pin GPIO Raspberry Pi 3
(Sumber: Raspberrypi-Spy.co.uk, 2017)



Gambar 2.6 Skematik Raspberry Pi 3

(Sumber: *elinux.org*, 2012)

Raspberry pi adalah sebuah komputer berukuran cukup kecil. dimana ukurannya hanya sebesar kartu kredit. Seperti yang digambarkan pada gambar 2.4. Tujuan awal dibuatnya *Raspberry pi* adalah untuk tujuan edukasi. *Raspberry pi* dapat dihubungkan ke monitor komputer maupun televisi dan menggunakan *keyboard* dan *mouse* standar. Perangkat ini dapat digunakan oleh orang-orang dari segala usia untuk mengeksplorasi dunia komputer dan belajar tentang bahasa pemrograman seperti *Scratch* dan *Python*.

Raspberry pi amat sangat disukai oleh para *developer* elektronik yang proyeknya membutuhkan lebih dari mikrokontroller dasar. Hal itu dikarenakan ukuran *Raspberry pi* yang kecil dan memiliki harga yang cukup terjangkau (*Raspberrypi.org*).

Pada penelitian ini Peneliti menggunakan *Raspberry pi* dengan 3 model B. Model ini memiliki RAM sebesar 1 GB dan untuk CPU menggunakan *processor Broadcom Quad Core 1.2 GHz*. *Raspberry pi 3* model b ini juga dilengkapi dengan *HDMI port*, *4 USB ports*, *ethernet port*, empat puluh *pin GPIO*. seperti yang ditunjukkan oleh gambar 2.2. untuk dapat menggunakan pin GPIO yaitu dengan menggunakan library *Rpi.GPIO* yang sudah terinstall dalam *raspberry pi*. *Micro SD Card slot* dan lain lain. *Raspberry pi 3* ini juga identik dengan generasi sebelumnya yaitu *Raspberry pi 2* dan kompatibel dengan *Raspberry pi 1* (*Raspberry Pi 3 Model B*. 2017).

Tabel 2.2 Spesifikasi *Raspberry Pi 3*

SoC	Broadcom BCM2837
CPU	4x ARM Cortex-A53. 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
Jaringan	10/100 Ethernet. 2.4GHz 802.11n wireless
Bluetooth	Bluetooth 4.1 Classic. Bluetooth Low Energy
Penyimpanan	microSD
PIN GPIO	40-pin header. Populated
PORTS	HDMI. 3.5mm analogue audio-video jack. 4x USB 2.0. Ethernet. Camera Serial Interface (CSI). Display Serial Interface (DSI)

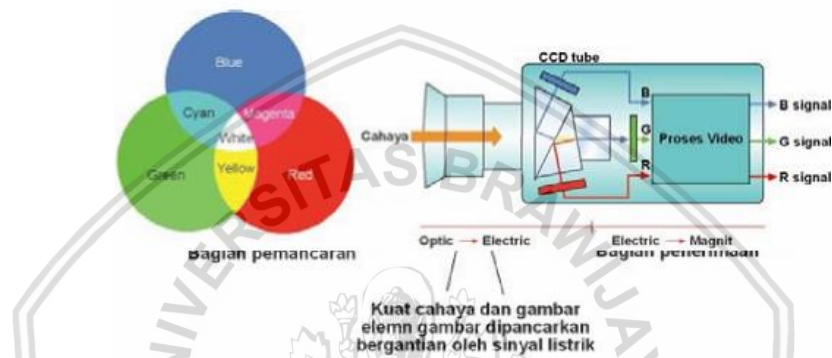
Sumber: (*raspberrypi.org*, 2017)

2.2.5 Webcam Logitech C270



Gambar 2.7 Webcam Logitech C270

Sumber: (Logitech.com, 2014)



Gambar 2.8 Prinsip Kerja Kamera

Sumber: (Dasar Fotografi Cara Cepat Memahami Fotografi, 2017)

Istilah *webcam* merupakan sebuah akronim dari *Web Camera*. *Webcam* adalah kamera digital yang dapat terhubung secara langsung ke komputer. Sesuai dengan namanya hasil rekaman dari kamera ini bisa langsung diakses melalui *world wide web* (www). Fungsi dasar dari penggunaan webcam yaitu sebagai pengambil gambar, perekam video dan lain-lain. *Webcamera* memiliki berbagai macam jenis, ada yang ditancapkan secara langsung di port usb pada computer dan ada juga yang dihubungkan melalui *wireless*. Secara fisik, sebuah *webcam* terdiri dari beberapa *elemen* yaitu lensa standar, papan sirkuit yang berfungsi untuk menangkap sinyal gambar, *casing* untuk menutupi lensa standar, kabel *support* sebagai pengontrol ketinggian dan sudut pandang *webcam* (Laksamana Media, 2009)

Dalam penelitian ini, peneliti menggunakan jenis *Webcam* Logitech c270. Fitur yang dimiliki *webcam* ini yaitu dapat merekam video dengan kualitas HD 720p, kamera ini mempunyai resolusi 3MP dengan mikrofon yang sudah termasuk di dalamnya. Webcam Logitech c270 juga mempunyai fitur *automatic light correction* dan *noise reduction*. Dan memiliki *Port usb* dengan tipe *usb 2.0*.

Tabel 2.3 Spesifikasi webcam Logitech 270

Connection Type	<i>Corded USB</i>
USB Type	<i>High Speed USB 2.0</i>
USB VID_PID	<i>VID_046D&PID_081A</i>
Microphone	<i>Built-in. Noise Supression</i>
Lens and Sensor Type	<i>Plastic</i>
Focus Type	<i>Fixed</i>
Field of View (FOV)	<i>60°</i>
Focal Length	<i>4.0 mm</i>
Optical Resolution (True)	<i>1280 x 960 1.2MP</i>
Image Capture (4:3 SD)	<i>320x240. 640x480 1.2 MP. 3.0 MP</i>
Image Capture (16:9 W)	<i>360p. 480p. 720p</i>
Video Capture (4:3 SD)	<i>320x240. 640x480. 800x600</i>
Video Capture (16:9 W)	<i>360p. 480p. 720p.</i>
Frame Rate (max)	<i>30fps @ 640x480</i>
Video Effects (VFX)	<i>N/A</i>
Right Light	<i>Right Light 2</i>
Buttons	<i>Other NA</i>
Indicator Lights (LED)	<i>Activity/Power</i>
Privacy Shade	<i>No</i>
Clip Size (max)	<i>0 to infinity</i>
Cable Length	<i>5 Feet or 1.5 Meters</i>

Sumber: (www.logitech.com, 2014)

2.2.6 OpenCV



Gambar 2.9 OpenCV

(Sumber : opencv.org, 2011)

OpenCV adalah sebuah *library* gratis atau *open source* dan bisa didownload oleh siapa saja yang digunakan untuk mengolah gambar dan video. CV (*Computer*

Vision) pada kata *OpenCV* maksudnya adalah komputer yang digunakan untuk mengolah image (citra/gambar) yang ditangkap oleh alat perekam. seperti kamera atau *webcam* yang kemudian dikonversi dari analog ke digital dan diolah di dalam komputer. Ada banyak tujuan dari pengolahan *image*. seperti untuk memperbaiki kualitas gambar atau untuk identifikasi gambar (Tabloid Reformata Edisi 106 Mei)

OpenCV dibuat dengan tujuan untuk menyediakan infrastruktur umum untuk aplikasi visi komputer dan juga dengan tujuan mempercepat penggunaan persepsi mesin dalam produk komersial menjadi produk berlisensi BSD. *OpenCV* memudahkan bisnis dalam memanfaatkan dan memodifikasi kode. *OpenCV* merupakan *open source computer vision library* untuk bahasa pemrograman C/C++. dan telah dikembangkan ke *python*, *java*, *matlab*. *OpenCV* dibuat oleh *intel* dan sekarang didukung oleh Willow Garage dan *Itseez*. *Intel* meluncurkan versi pertama dari *OpenCV* pada 1999. dan awalnya memerlukan *library* dari *Intel Image Processing Library*. *OpenCV* mendukung *multiplatform*. dapat mendukung baik *windows* ataupun *linux*. dan sekarang telah mendukung *MacOSX* dan *android* Saat ini versi yang terbaru adalah 3.4.1 dengan tanggal rilis 27 Februari 2018. (*OpenCV.org*. 2011).

OpenCV bersandar sebagian besar terhadap aplikasi visi *real-time* dan mengambil keuntungan dari instruksi *MMX* dan *SSE* jika tersedia. *CUDA* dan *OpenCL* antarmuka berfitur lengkap sedang dikembangkan secara aktif saat ini. Ada lebih dari 500 *algoritma* dan sekitar 10 kali lebih banyak fungsi yang menyusun atau mendukung *algoritma* tersebut. *OpenCV* ditulis secara asli di *C++* dan memiliki antarmuka *templated* yang bekerja dengan mulus dengan *kontainer STL*.

Contoh penggunaan *library* dengan *open cv* dalam penelitian ini yaitu pada *webcam* dengan *library* menggunakan fungsi *cam=cv2.VideoCapture()* dan *cam.read()* pada *openCV*. *cv2.VideoCapture* berfungsi sebagai inisialisasi penggunaan *webcam* dan *cam.read* digunakan untuk membaca frame yang ditangkap *webcam* secara terus menerus, kemudian pada penggunaan *Template Matching* dapat menggunakan fungsi *cv2.MatchTemplate* dan *CV_TM_CCOEFF_NORMED* yang terdapat didalam *openCV*.

2.2.7 Template Matching

Metode *Template Matching* adalah sebuah Metode algoritma pengenalan citra yang berfungsi untuk mengenali bagian – bagian dari citra (Bahri. 2012). Cara yang digunakan pada metode ini adalah dengan membandingkan nilai dari tiap piksel pada gambar asli dengan nilai tiap piksel pada *template* gambar yang telah disimpan sebelumnya. Proses pencocokannya dilakukan dengan melihat nilai kemiripan dan nilai batas ambang dari kedua citra. Bila kedua citra memiliki jumlah nilai kemiripan diatas ataupun sama dengan nilai batas ambang yang telah ditentukan maka kedua citra akan dikategorikan sebagai objek yang sama.

Kelebihan dalam menggunakan metode *template Matching* ini adalah tidak memakan waktu yang banyak dalam pemrosesannya. Hal ini dikarenakan proses *template matching* yang menggunakan proses berupa perhitungan *matriks*. Kekurangan dari metode *Template Matching* adalah metode ini membutuhkan template yang cukup banyak supaya hasil dari pengklasifikasian semakin akurat dan optimal. Berikut ini beberapa Method Template Matching yang disediakan openCV. Proses pencocokan dilakukan secara berulang – ulang sebanyak template yang disiapkan.

Tabel 2.4 Method Template Matching yang disediakan OpenCV

No	Method	Rumus
1	CV_TM_SQDIFF	$R(x, y) = \sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))^2$
2	CV_TM_SQDIFF_NORMED	$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$
3	CV_TM_CCORR	$R(x, y) = \sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))$
4	CV_TM_CCORR_NORMED	$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$
5	CV_TM_CCOEFF	$R(x, y) = \sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))$
6	CV_TM_CCOEFF_NORMED	$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$

Pada tabel 2.4 OpenCV menyediakan *Method Template Matching* sebanyak 6. Penelitian ini menggunakan metode CV_TM_CCOEFF_NORMED karena metode ini yang paling sering digunakan orang dan hasil keluaran yang familiar yaitu 0 - 1 dengan 1 sebagai nilai dengan kecocokan paling tinggi.

Jika *Image* (I) = $W \cdot H$ dan *Template* (T) = $w \cdot h$,

maka result (R) = $(W - w + 1) \cdot (H - h + 1)$

$$T'(x', y') = T(x', y') - \frac{1}{w \cdot h} \cdot \sum_{x'', y''} I(x + x'', y + y'')$$

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{w \cdot h} \cdot \sum_{x'', y''} I(x + x'', y + y'')$$

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}} \quad (2.1)$$

Keterangan :

T = *Template*

I = *Image*

x = Nilai x pada matriks Gambar Obyek

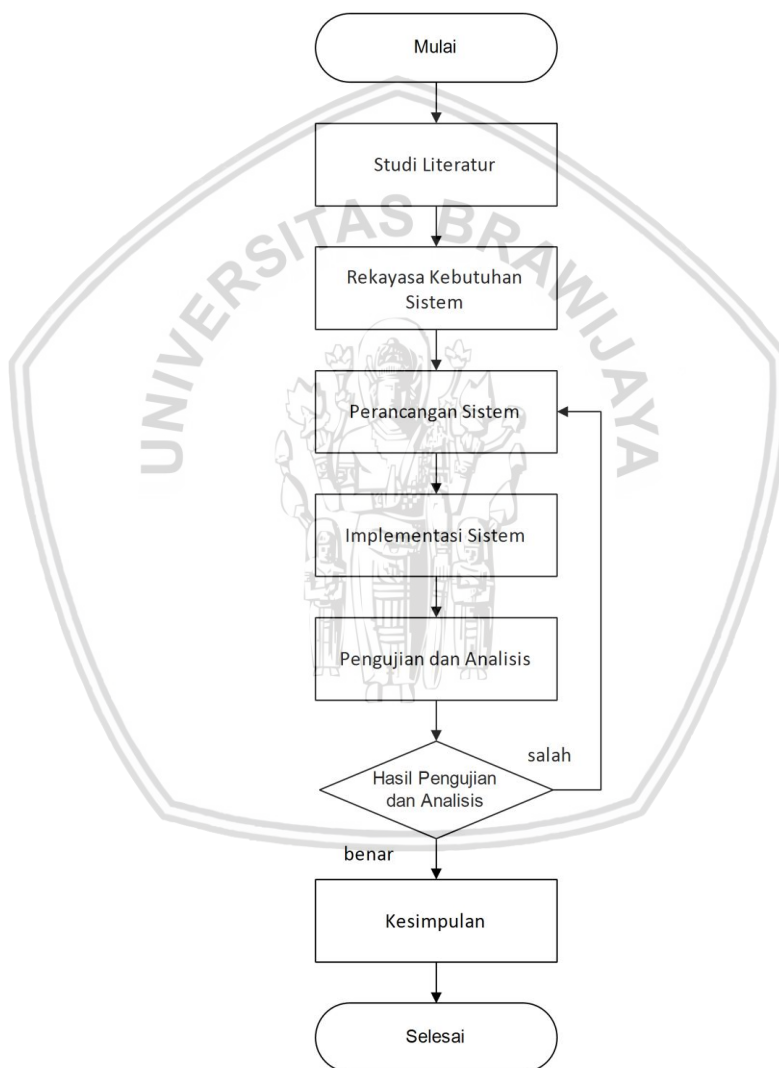
x' = Nilai x pada matriks Gambar Template
 y = Nilai y pada matriks Gambar Obyek
 y' = Nilai y pada matriks Gambar template

Jika tingkat kemiripan diatas nilai *threshold* yang ditentukan maka citra yang diambil bisa dikatakan sesuai dengan templatnya.



BAB 3 METODE PENELITIAN

Penelitian yang Penulis lakukan tergolong dalam penelitian implementatif. Dalam melakukan penelitian ini terlebih dahulu mencari studi literatur yang berkaitan dengan *Template Matching*. Selanjutnya dilakukan analisis kebutuhan sistem. Kemudian sistem akan dirancang, diimplementasikan serta diuji. Supaya lebih jelas, penulis membuat diagram alir yang menggambarkan bagaimana metodologi dari penelitian dilakukan.



Gambar 3.1 Diagram Alur Metodologi Penelitian

3.1 Studi Literatur

Dalam penelitian ini, peneliti menggunakan berbagai teori yang diambil dari beberapa referensi, meliputi:

3.1.1 Teori terkait *webcam Logitech C270*

Melakukan kajian terhadap *webcam Logitech C270* yang spesifikasinya mampu untuk menangkap gambar dari objek dihadapannya dengan baik sehingga hasil penelitian pun akan lebih akurat serta juga lebih baik lagi.

3.1.2 Teori terkait *Raspberry Pi 3*

Melakukan kajian terhadap *Raspberry Pi 3* terutama yang berkaitan dengan pengambilan gambar terhadap objek .

3.1.3 Teori terkait *Template Matching*

Melakukan kajian terkait apa itu *Template Matching* dan bagaimana penggunaan metode ini dapat berpengaruh terhadap hasil dari sistem jenis buah mangga.

3.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk menganalisis kebutuhan sistem pada penelitian yang dilakukan, sehingga sistem dapat bekerja sesuai dengan harapan. Berikut ini adalah kebutuhan fungsional yang dibutuhkan oleh sistem dalam penelitian ini:

1. Kamera pada sistem dapat mengambil gambar objek buah mangga
2. Gambar dapat diolah untuk diklasifikasikan menggunakan metode *Template Matching*
3. Sistem dapat menampilkan hasil klasifikasi jenis buah manga.

3.2.1 Kebutuhan Perangkat Keras

Berikut ini beberapa kebutuhan perangkat keras yang akan digunakan dalam pembuatan sistem pada penelitian ini:

1. Raspberry Pi 3
2. Kamera Logitech C270
3. Kabel Jumper
4. LED
5. Resistor 330 ohm
6. Push Button
7. Laptop

3.2.2 Kebutuhan Perangkat Lunak

Bagian ini berisi tentang analisa terkait kebutuhan perangkat lunak yang dibutuhkan system dalam penelitian ini, Perangkat lunak yang dibutuhkan adalah Open CV dan Python 3.

3.3 Perancangan

Pada sub bab ini. sistem dirancang dengan terlebih dahulu menghubungkan kamera dengan *raspberry pi* melalui *pin* yang telah ditentukan. Kemudian data dari sensor akan diproses oleh *raspberry pi*. Hasil dari proses ini yakni nama dari jenis buah mangga.

Bagan dibawah ini akan menunjukkan bagaimana perancangan sistem yang saya buat.



Gambar 3.2 Diagram blok Perancangan Sistem

3.4 Implementasi Sistem

Pada tahap implementasi sistem, sistem yang telah dirancang akan diterapkan atau diimplementasikan secara langsung. Buah mangga diletakkan di bawah *webcam*, selanjutnya sistem mengambil data yang didapat dari *webcam* yang kemudian di proses menggunakan algoritma *Template matching*. Selanjutnya sistem akan mengklasifikasikan jenis dari buah mangga tersebut.

3.5 Pengujian dan Analisis

Pada tahap pengujian dan analisis, sistem yang sudah jadi akan diimplementasikan dan diuji kemudian dianalisis. Pengujian dilakukan dengan cara mengukur sistem dapat mendeteksi jenis buah mangga dengan akurat, serta kecepatan waktu pemrosesan sistem dalam melakukan proses klasifikasi.

Setelah diuji kemudian akan dilakukan analisis mengenai kekurangan yang terjadi pada sistem apa yang mengakibatkan sistem tersebut melakukan kesalahan pengklasifikasian. Proses analisis berguna terhadap sistem yang dibuat, setelah kekurangan diketahui sistem dapat dibenahi kedepannya agar dapat berjalan dengan lebih baik

3.6 Kesimpulan

Tahap terakhir adalah tahap kesimpulan. Pada tahap ini akan dipaparkan kesimpulan dari penelitian yang telah dilakukan, apakah telah sesuai dengan rumusan masalah atau belum. Pada tahap kesimpulan juga diambil beberapa

saran yang dibuat dengan tujuan untuk memperbaiki penelitian ini dan sebagai dasar atau pertimbangan untuk peneliti lain yang akan melakukan penelitian serupa.



BAB 4 REKAYASA KEBUTUHAN

Bab ini akan dijelaskan secara rinci mengenai gambaran umum sistem yang akan dibuat. Analisis kebutuhan fungsional, kebutuhan perangkat keras dan kebutuhan perangkat lunak serta batasan desain sistem.

4.1 Gambaran Umum Sistem

Sistem Pendeteksi Jenis Buah Mangga Menggunakan *Template Matching* Berbasis *Raspberry Pi* merupakan suatu system yang dapat menentukan suatu jenis buah mangga dengan parameter berupa warna dan bentuk pada buah. Bentuk dan warna ini akan dibaca dengan *webcam C270*, dan akan diproses ke *raspberry pi 3* model B Berdasarkan nilai dari kedua parameter tersebut maka dapat dilakukan klasifikasi hasilnya dengan menggunakan metode *Template Matching*. Sistem ini membantu menyelesaikan permasalahan penentuan tingkat suatu jenis buah mangga secara otomatis. Selain itu penggunaan metode *Template Matching* menjadi metode yang tepat karena dapat menghasilkan tingkat akurasi yang tinggi sesuai dengan peluang jumlah fakta yang dianggap benar berdasarkan data yang sebenarnya. Hasil dari pengolahan sistem ini secara otomatis akan ditampilkan pada layar monitor ataupun *LED*.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengetahui semua kebutuhan yang diperlukan untuk Sistem Pendeteksi Jenis Buah Mangga. Analisis kebutuhan sistem dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non fungsional.

1.2.1 Kebutuhan Fungsional

Berikut ini adalah kebutuhan fungsional yang harus mampu dilakukan oleh sistem:

1. Kamera *Webcam C270* untuk membaca atau mengambil data yang dideteksi. Kamera *Webcam c270* bertugas untuk membaca tampilan fisik mangga yang kemudian dapat dideteksi oleh sistem mengenai jenis buah mangga.
2. *Raspberry Pi 3* Model B untuk mengolah citranya data dari *webcam* untuk mengklasifikasikan hasilnya menjadi jenis mangga yang berbeda berdasarkan metode *Template Matching*.
Pada fungsionalitas ini bertugas untuk melakukan proses klasifikasi dengan menggunakan metode *Template Matching* berdasarkan *input* yang diperoleh dari pembacaan *webcam* dan *raspberry pi* serta dari data latih yang ditanamkan pada pemrograman *Template Matching* tersendiri.
3. Sistem dapat menampilkan hasil klasifikasi jenis buah Mangga pada monitor.

Pada fungsionalitas ini bertugas untuk dapat menampilkan hasil *output* dari sistem yaitu hasil klasifikasi jenis buah pada monitor komputer dan *LED*.

4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional dari sistem ini terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak yang dijelaskan secara rinci dibawah ini.

4.2.2.1 Kebutuhan Perangkat Keras

Guna mendukung implementasi pembuatan sistem dari sisi perangkat keras maka di perlukan beberapa alat yang dijelaskan berikut ini:

1. *Raspberry Pi 3 Model B*

Penelitian ini menggunakan Raspberry Pi dikarenakan Raspberry memiliki spesifikasi yang cukup memadai untuk memproses citra digital dengan menggunakan OpenCV.

2. Kamera *Webcam Logitech C270*

Kamera Webcam Logitech C270 digunakan dalam penelitian ini karena Kamera ini memiliki kemampuan yang cukup baik dalam mengambil gambar.

3. Laptop

Laptop dalam hal ini berfungsi sebagai media untuk membuat program untuk mikrokontroler selain itu berfungsi sumber daya dari sistem yang akan dibuat. Adapun spesifikasi laptop yang digunakan antara lain:

- Model perangkat: ASUS X450JN
- *Prosesor*: Intel Core i7 4700HQ
- RAM: 8 GB
- Sistem Operasi: Windows 10 Pro 64bit

4. *LED*

LED digunakan pada sistem untuk menampilkan hasil klasifikasi jenis buah Mangga. Pada penelitian ini menggunakan *LED* karena *LED* memiliki harga yang terjangkau.

5. Tombol *Push Button*

Tombol ini berfungsi untuk memulai sistem berjalan, dan digunakan karena penggunaannya yang mudah.

6. Resistor 330 ohm

Resistor yang digunakan memiliki nilai resistansi sebesar 330 ohm yang sudah cukup untuk menyalakan satu buah LED

4.2.2. Kebutuhan Perangkat Lunak

Pada bagian ini terdapat kebutuhan perangkat lunak yang dibutuhkan oleh sistem yaitu:

1. *OS Raspbian-Stretch*

Penelitian ini menggunakan *OS Raspbian-Stretch* pada raspberry pi dikarenakan penggunaannya yang mudah dan *user friendly*.

2. *OpenCV*

Penelitian ini menggunakan *OpenCV* karena *OpenCV* merupakan *software* yang gratis.

3. *Python 3*

Digunakan karena memiliki bahasa pemrograman yang mudah

4.3 Batasan Desain Sistem

Dalam pembuatan Sistem Pendeteksi jenis buah mangga ini terdapat beberapa batasan sehingga lingkup pembahasan, perancangan, maupun pengimplementasiannya tidak terlalu luas. Adapun batasan-batasan desain sistem ini adalah sebagai berikut:

1. Sistem melakukan klasifikasi terhadap 3 jenis buah manga, yaitu Manalagi, Arum Manis, dan Golek
2. Sebelum sistem digunakan, sistem harus diaktifkan terlebih dahulu untuk mengkalibrasi *Raspberry Pi* selama 10 menit.
3. Sistem hanya dapat berjalan ketika menerima *trigger* dari pengguna.
4. Sistem dirancang untuk sistem pencahayaan 200 - 300 lux.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

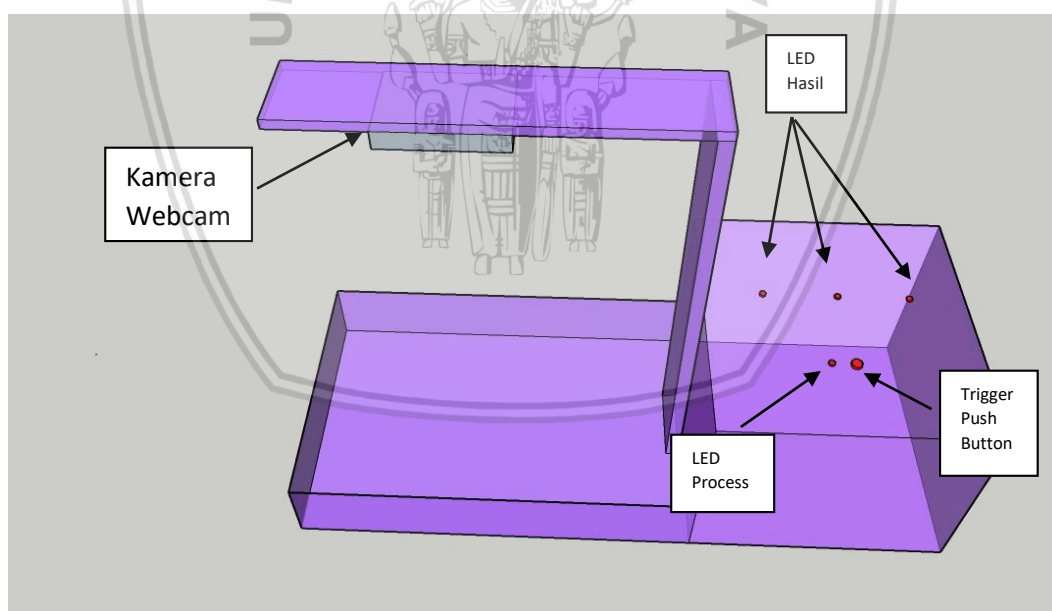
Bagian dari bab 5 ini menjelaskan mengenai proses perancangan dan implementasi sistem secara terperinci baik perancangan dan implementasi dari sisi perangkat keras maupun dari sisi perangkat lunak.

5.1 Perancangan Sistem

Dalam sub-bab ini akan menjabarkan cara perancangan sistem dimulai dari perancangan purwarupa alat, perancangan perangkat keras hingga perancangan perangkat lunak.

5.1.1 Perancangan Purwarupa Alat Pendeteksi Jenis Buah Mangga

Dalam melakukan desain purwarupa dari Pendeteksi Jenis Buah Mangga ini perlu diperhatikan peletakkan tiap-tiap komponen serta ukuran alat yang akan dikembangkan. Pembuatan desain sistem dirancang menggunakan aplikasi *Sketchup* untuk menggambarkan bentuk purwarupa yakni berupa kotak yang terbuat dari akrilik berwarna putih berukuran panjang 38 cm, lebar 20 cm dan tinggi 25 cm. Bentuk purwarupa alat ditunjukkan pada Gambar 5.1.



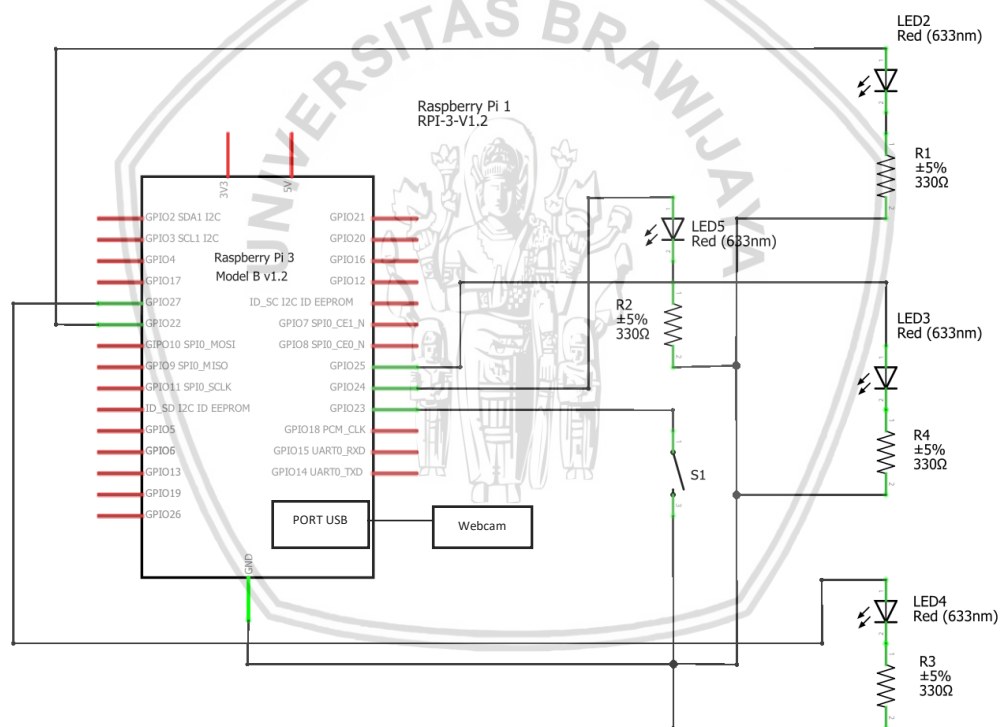
Gambar 5.1 Desain Sistem Pendeteksi Jenis Buah Mangga

Posisi kamera diletakkan di atas area tempat objek dan menghadap kebawah untuk menghindari interfensi cahaya lain selain warna dari objek yang akan dibaca dimana dalam hal ini adalah Mangga. Mangga diletakkan tepat dibawah kamera. LED sebanyak 4 buah ditempatkan diatas kotak tidak jauh dari letak peletakkan kamera untuk memudahkan pengguna melihat langsung hasil

olahan sistem. Selain itu di permukaan atas dari kotak juga diletakkan *push button* yang digunakan untuk men-*trigger* data sensor yang akan diolah.

5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras ini dilakukan atas dasar analisis kebutuhan perangkat keras dan spesifikasi dari masing-masing perangkat keras agar dapat membangun sistem yang diharapkan. Berdasarkan Gambar 3.2 yang merupakan blok diagram sistem maka pada tahap perancangan perangkat keras ini menjabarkan secara rinci hubungan skematik pin-pin yang digunakan antara tiap komponen perangkat keras. dimana dalam hal ini kamera yang merupakan input dihubungkan dengan *Raspberry Pi 3* sebagai pengolah data sehingga hasil olahan datanya dapat ditampilkan melalui Layar monitor dan *LED*. Skematik diagram perancangan perangkat keras Pendeteksi Jenis Buah Mangga ini ditunjukkan pada Gambar 5.2.



Gambar 5.2 Diagram Skematik Sistem

Pada diagram skematik diatas terlihat bahwa ada *push button* dengan jenis *pull up* yang fungsinya telah dijelaskan pada deskripsi sistem. Rangkaian *push button* salah satu kakinya dihubungkan dengan *pin GPIO 23* pada *Raspberry Pi* sedangkan pada kaki lainnya dibagi dengan menghubungkan ke *ground* dari *Raspberry*. Selanjutnya *webcam* dihubungkan dengan *raspberry* pada *port USB*. Selanjutnya rangkaian *LED* dihubungkan dengan resistor 330 ohm. Nilai 330 ohm didapat dari perhitungan dengan rumus $R = (V_S - V_L) / I$.

Dimana :

R = Nilai Resistor yang diperlukan (dalam Ohm (Ω))

V_s = Tegangan Input (dalam Volt (V))

V_L = Tegangan LED (dalam Volt (V))

I = Arus Maju LED (dalam Ampere (A))

Jika tegangan Input adalah dari pin GPIO adalah 5V dan LED yang digunakan adalah LED Merah 5 mm yang memiliki tegangan $V_L = 1.7V$. Arus Maju (I) adalah 20mA (diubah menjadi Ampere menjadi 0.02A). maka:

$$R = (V_s - V_L) / I$$

$$R = (5V - 1.7V) / 0.02A$$

$$R = 165\Omega$$

Led yang digunakan sistem membutuhkan resistor minimal sebesar 165 ohm. Setelah nilai didapatkan selanjutnya yaitu menghubungkan kaki LED dengan pin GPIO *Raspberry* yang dijelaskan pada Tabel 5.1.

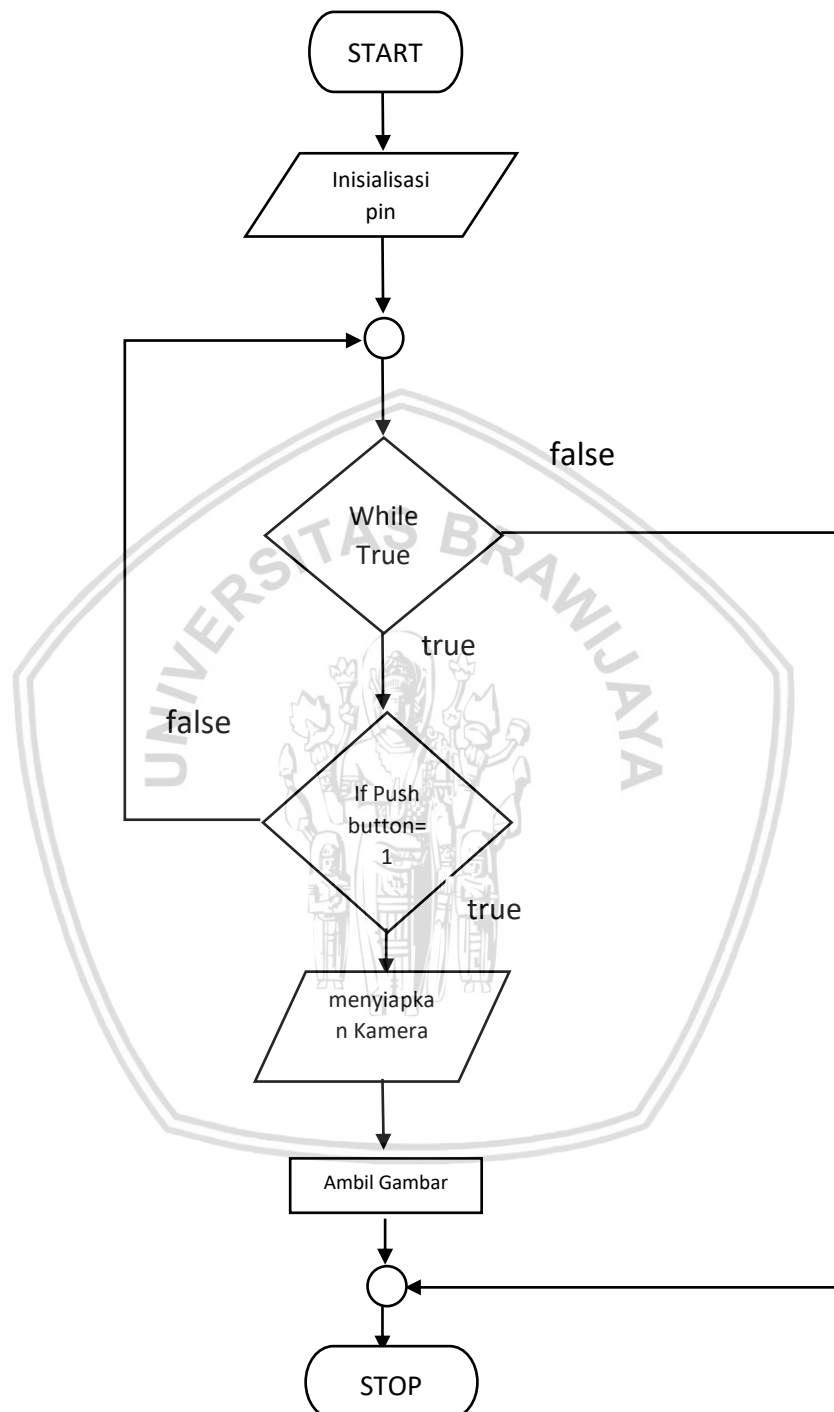
Tabel 5.1 Keterangan koneksi *pin LED* dengan *pin GPIO Raspberry*

<i>Kaki Anoda LED</i>	<i>PIN GPIO Raspberry Pi 3</i>
<i>Anoda LED1</i>	<i>GPIO 22</i>
<i>Anoda LED2</i>	<i>GPIO 25</i>
<i>Anoda LED3</i>	<i>GPIO 27</i>
<i>Anoda LED Indikator</i>	<i>GPIO 24</i>

5.1.3 Perancangan Perangkat Lunak

Pada subbab perancangan perangkat lunak dibagi menjadi 3 pembahasan. yakni perancangan perangkat lunak pada *mikrokontroler* untuk mengambil data gambar yang akan diolah. proses pengambilan template. serta perancangan perangkat lunak untuk melakukan proses klasifikasi dengan metode *Template Matching*.

5.1.3.1 Perancangan Pengambilan Data Kamera

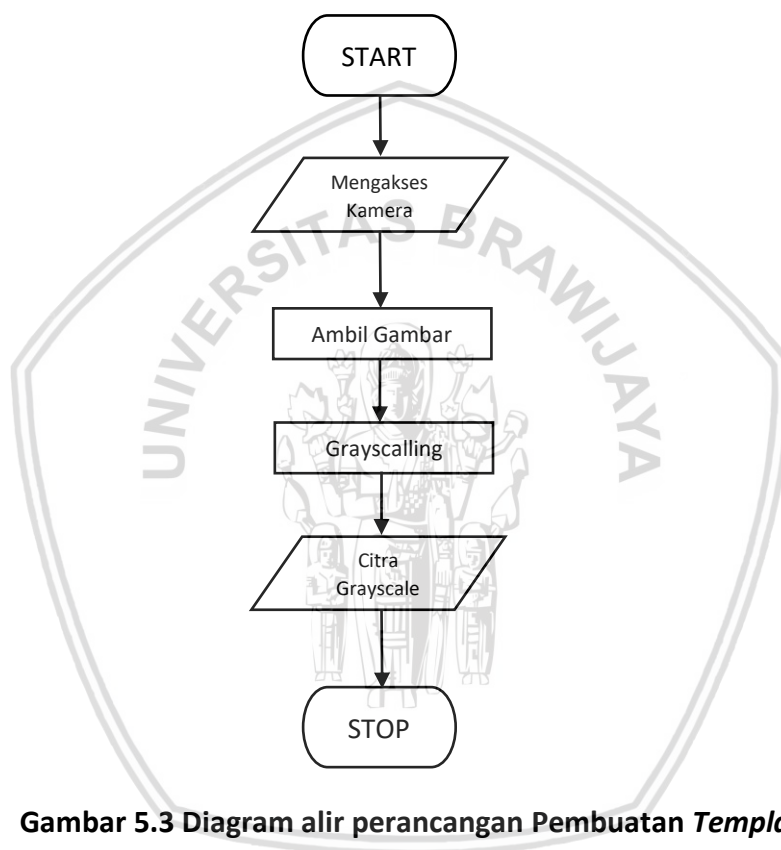


Gambar 5.3 Diagram alir perancangan perangkat lunak pengambilan gambar

Proses perancangan perangkat lunak untuk pengambilan gambar ditunjukkan pada Gambar 5.3 diatas. hal ini dimaksudkan untuk menentukan hasil pembacaan gambar dari kamera dimana yang akan diolah untuk dilakukan klasifikasi dengan menggunakan *Template Matching*. Sistem dimulai dengan

melakukan inisialisasi *pin* yang digunakan oleh LED dan *push button*. Selanjutnya setelah inisialisasi *pin raspberry* akan melakukan proses pengecekan penekanan *push button*. Apabila terjadi penekanan *push button*, raspberry akan mempersiapkan webcam untuk mengambil gambar. Jika sudah siap, maka kamera akan segera mengambil gambar dengan perintah `cv2.VideoCapture()` yang telah tersedia dalam openCV. Setelah gambar didapat maka gambar siap untuk diproses untuk mengetahui klasifikasi jenis buah manga dengan menggunakan metode *Template Matching*.

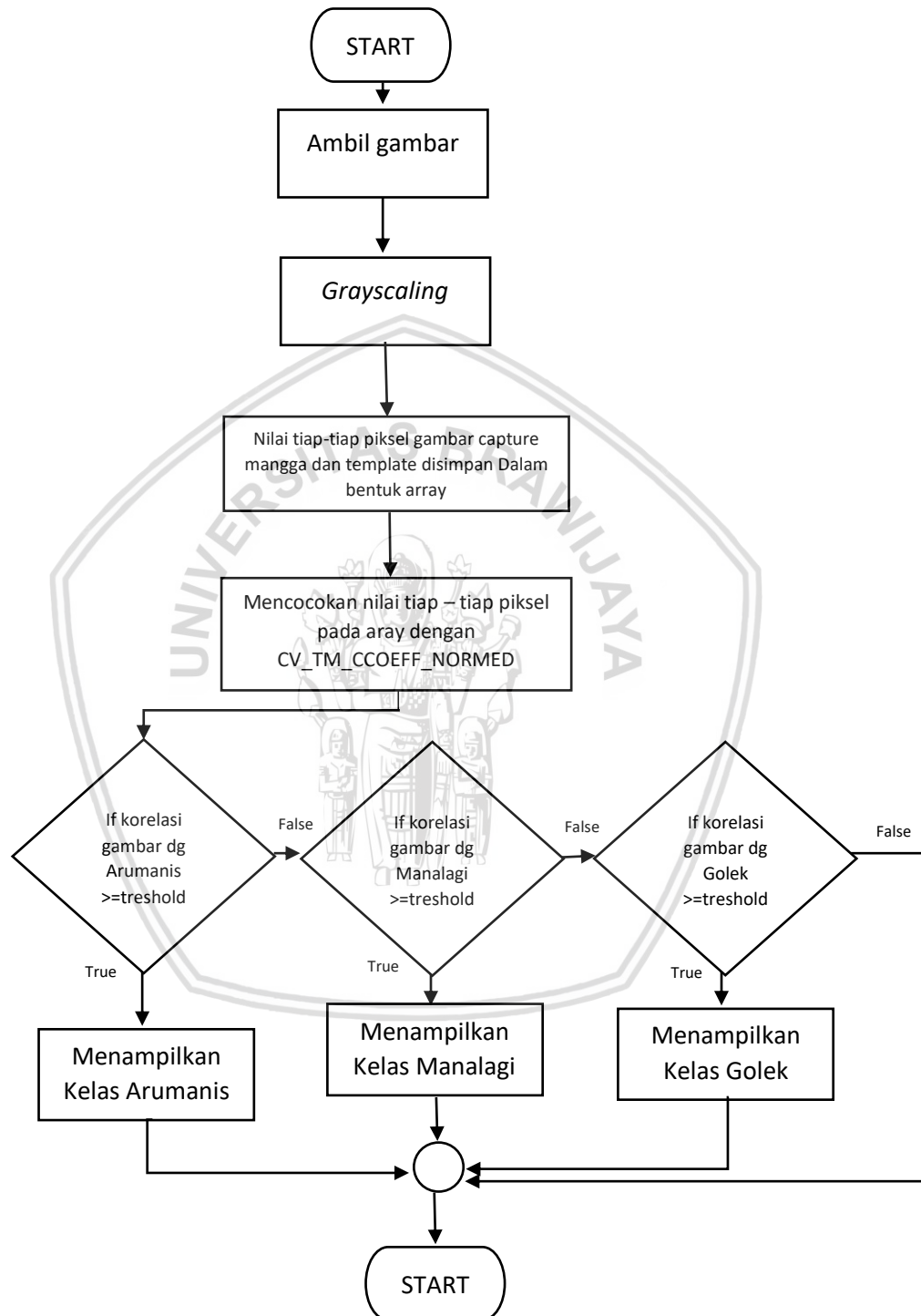
5.1.3.2 Perancangan Pembuatan *Template*



Gambar 5.3 Diagram alir perancangan Pembuatan *Template*

Pada Gambar 5.3 merupakan proses pembuatan *template*. Proses pembuatan template dimulai dengan mengakses kamera dengan menggunakan perintah `cv2.VideoCapture()` yang ada dalam openCV kemudian kamera mengambil citra manga. Setelah gambar didapat lalu citra diubah dari *RGB* dan *grayscale* dengan perintah `cv2.cvtColor(gambar, cv2.COLOR_BGR2GRAY)`. Grayscale dipilih karena library *template matching* menggunakan openCV hanya membutuhkan satu jenis nilai dari tiap – tiap piksel. sedangkan *RGB* memiliki tiga komponen dalam 1 pikselnya. Kemudian citra *RGB* yang sudah di ubah menjadi *grayscale* akan disimpan sebagai database *template* dengan menggunakan perintah `cv2.imwrite()`.

5.1.3.3 Perancangan Pengenalan Klasifikasi Dengan *Template Matching*



Gambar 5.4 Diagram alir Perancangan Pengenalan Klasifikasi Dengan *Template Matching*

Pada Gambar 5.4 merupakan proses pengenalan klasifikasi buah mangga dengan *template matching*. Kamera akan mengambil gambar mangga yang telah disiapkan. kemudian setelah gambar diambil. maka gambar akan diubah dari RGB ke gray dengan perintah `cv2.COLOR_BGR2GRAY`. Citra mangga yang telah diubah dari *RGB* ke *Grayscale* tiap nilai pada pikselnya akan dimasukkan ke dalam bentuk array dengan perintah `a.b= gambar.shape[::]`. Selanjutnya nilai array gambar akan dicocokkan dengan *template* yang telah dikumpulkan sebelumnya dengan menggunakan *template matching*. Proses pencocokan dilakukan secara berulang – ulang sesuai dengan banyaknya *template* yang disiapkan. Setiap *template* yang dicocokkan akan dihitung kecocokannya terhadap citra *grayscale* yang diambil dari kamera dengan cara membandingkan nilai koefisien korelasi dari gambar yang di *capture* dengan *templat*nya. Untuk menghitung kecocokan tiap piksel dapat menggunakan fungsi `cv2.matchTemplate(namagambar, namatemplate, cv2.TM_CCOEFF_NORMED)` parameter `cv2.TM_CCOEFF_NORMED` merupakan salah satu metode *template matching* yang disediakan oleh openCV dengan perhitungan yang telah dijelaskan di subbab 2.7. jika nilai koefisien sudah didapatkan maka kita akan membandingkan nilai tersebut dengan nilai threshold yang kita dapatkan dari proses trial dari masing – masing kelas sampai menemukan hasil kasifikasi yang sempurna. Nilai threshold yang digunakan dalam sistem ini sebesar 0,78. Membandingkan nilai dilakukan secara berurutan dari kelas A. kelas B. kelas C. Jika nilai korelasi gambar dengan kelas A lebih besar dari nilai threshold maka sistem akan menampilkan kelas A. jika tidak maka lanjut membandingkan ke kelas B. Jika nilai korelasi gambar dengan kelas B lebih besar dari nilai threshold maka sistem akan menampilkan kelas B. jika tidak maka lanjut membandingkan ke kelas C. Jika nilai korelasi gambar dengan kelas C lebih besar dari nilai threshold maka sistem akan menampilkan kelas C. jika tidak maka sistem akan berhenti.

5.2 Implementasi Sistem

Implementasi sistem merupakan tahap untuk merealisasikan pembuatan sistem berdasarkan semua perancangan yang telah dilakukan sebelumnya. Pada subbab ini menjelaskan satu per satu secara rinci terkait implementasi purwarupa. implementasi perangkat keras serta implementasi perangkat lunak.

5.2.1 Implementasi Alat Pendeteksi Jenis Buah Mangga

Dalam mengimplementasikan purwarupa alat pendeteksi jenis ini mengacu pada perancangan di subbab 5.1.1 yakni menggunakan papan akrilik dengan ketebalan 3 mm berwarna putih berukuran panjang 38 cm. lebar 20 cm

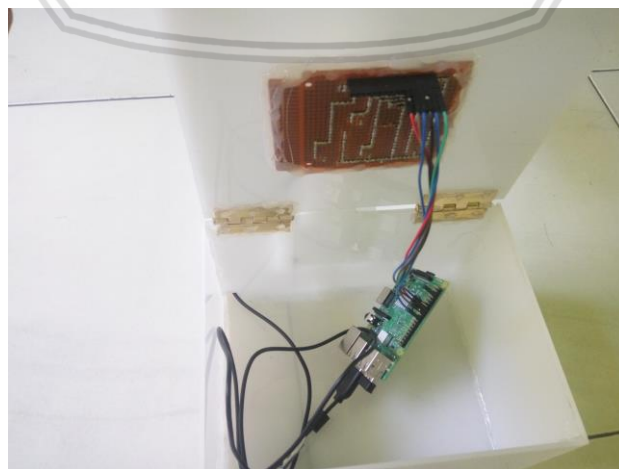
dan tinggi 30 cm. Hasil implementasi purwarupa beserta peletakan komponen elektronik yang digunakan ditunjukkan pada Gambar 5.5.



Gambar 5.5 Implementasi Purwarupa Alat Pendeteksi Jenis Buah Mangga

5.2.2 Implementasi Perangkat Keras

Tahap ini menjelaskan proses pengimplementasian perangkat keras yang mencakup komponen elektronik antara lain *Raspberry Pi*, *Webcam Logitech C270*, *LED*, dan *push button*. Keseluruhan komponen dirangkai menjadi satu dengan menggunakan *PCB* yang dihubungkan berdasarkan perancangan yang telah dijelaskan pada subbab 5.1.2



Gambar 5.6 Implementasi Rangkaian Webcam dan Raspberry Pi

Pada Gambar 5.6 menunjukkan hasil implementasi dimana *webcam Logitech C270* yang telah dirangkai sesuai dengan perancangan diletakkan dibagian dalam dari kotak seperti gambar dan dihubungkan ke *raspberry pi* melalui *port USB*.



Gambar 5.7 Implementasi Rangkaian LED dan Push Button

Berdasarkan perancangan perangkat keras *LED* dan *push button* yang telah dijelaskan pada subbab 5.1.2. hasil dari implementasinya ditunjukkan pada Gambar 5.7 diatas.

5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan proses realisasi program untuk Sistem Pendeteksi Jenis Buah Mangga ini berdasarkan perancangan yang telah dilakukan pada subbab 5.1.3. Dalam melakukan implementasi perangkat lunak ini sepenuhnya proses pengkodean program dilakukan pada Bahasa pemrograman *python 3.3* dimana diawal program dilakukan inisialisasi *library* yang digunakan untuk mempermudah pemrograman beberapa fungsi tertentu. Pada Tabel 5.2 ditunjukkan pengimplementasian *library* pada sistem ini. diantaranya adalah *library "RPi.GPIO"* untuk penggunaan *pin GPIO* pada *Raspberry Pi*. *library "time"* untuk penggunaan *delay* pada pemrograman. *library "cv2"* untuk melakukan pemrosesan citra. *library "os"* untuk penggunaan *command* seperti pada terminal *linux* dan *library "numpy"* untuk proses perhitungan matematika dengan menggunakan *matriks*.

Tabel 5.2 Kode Pemrograman Inisialisasi *Library* Sistem Pendeteksi Jenis Buah Mangga

Baris	Kode Program
1	<code>import RPi.GPIO as GPIO</code>
2	<code>import time</code>
3	<code>import cv2</code>
4	<code>import os</code>
5	<code>import numpy as np</code>

5.2.3.1 Implementasi Kode Program Inisialisasi *Pin GPIO*

Pada tahap implementasi kode program *pin GPIO* ini bertujuan untuk merealisasikan kode program inisialisasi pin - pin *GPIO Raspberry Pi* yang digunakan oleh sistem. Pada tabel 5.3 menjelaskan konfigurasi pin yang digunakan sistem pada *raspberry pi* seperti yang telah dijelaskan pada subbab – subbab sebelumnya. Baris ke-1 merupakan tipe penomoran *pin GPIO* berdasarkan *Broadcom SOC channel* seperti pada gambar 2.2. baris ke-2 *Pin GPIO 23* sebagai input *push button* yang akan digunakan dengan jenis pull up. baris ke-3 sampai 6 merupakan *Output dari LED*. dan baris ke-7 merupakan pengabaian bila terjadi error akibat penggunaan *Pin* yang telah dipakai sebelumnya.

Tabel 5.3 Kode Pemrograman Inisialisasi Pin GPIO

Baris	Kode Program
1	<code>GPIO.setmode(GPIO.BCM)</code>
2	<code>GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_UP)</code>
3	<code>GPIO.setup(24, GPIO.OUT)</code>
4	<code>GPIO.setup(25, GPIO.OUT)</code>
5	<code>GPIO.setup(22, GPIO.OUT)</code>
6	<code>GPIO.setup(27, GPIO.OUT)</code>
7	<code>GPIO.setwarnings(False)</code>

5.2.3.2 Implementasi Kode Program Pengecekan *Trigger* dan Pengambilan Gambar

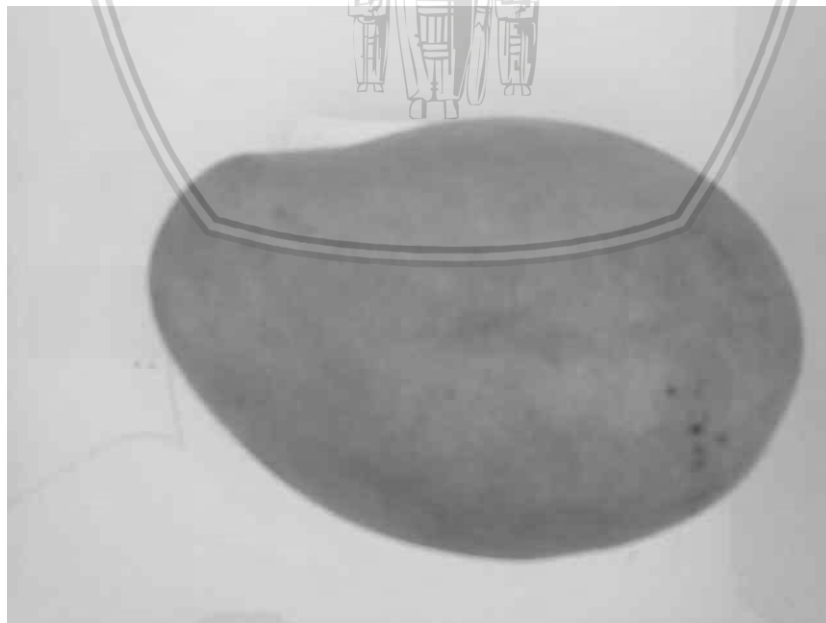
Pada tahap implementasi kode program *pin GPIO* ini bertujuan untuk merealisasikan kode pengecekan *trigger* dan pengambilan gambar. Pada tabel 5.4 menjelaskan pengambilan gambar melalui *webcam* dapat menggunakan fungsi `cv2.VideoCapture()` yang telah disediakan oleh *library opencv*. Pada baris ke-1 merupakan perintah untuk menjalankan secara terus menerus sampai ada *interrupt* dari pengguna. Pada baris ke-2 Jika *push button* di tekan maka akan menjalankan perintah dibawahnya yaitu mengambil gambar dan disimpan dengan nama “img2” pada direktori *home/pi*. Setelah itu pada baris ke-8 gambar akan diubah dari citra *RGB* ke citra *Gray* dengan menggunakan fungsi

`cv2.cvtColor(gambar, cv2.COLOR_BGR2GRAY)` yang telah disediakan *opencv*. Pada baris ke 9 menampilkan hasil pengambilan gambar ke layar monitor.



Gambar 5.8 Hasil Gambar *RGB*

Pada Gambar 5.8 merupakan hasil gambar yang telah diambil menggunakan *webcam Logitech C270*. Gambar tersebut masih bercitra *RGB* sehingga perlu diubah ke *gray* agar dapat diproses dengan menggunakan *template matching*



Gambar 5.9 Hasil Gambar *Grayscale*

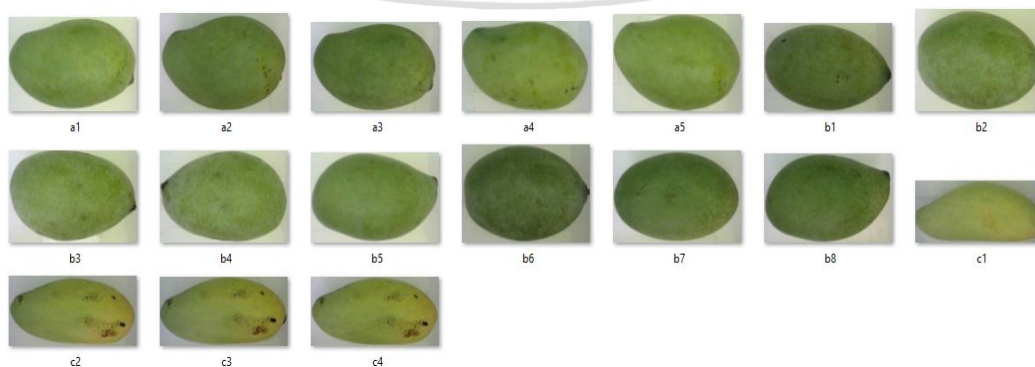
Pada Gambar 5.9 merupakan hasil gambar yang telah diubah dari citra *RGB* ke *Gray*. Gambar sudah siap untuk diproses ke tahap selanjutnya.

Tabel 5.4 Kode Pemrograman Pengecekan *Trigger* dan Pengambilan gambar

Baris	Kode Program
1	<code>while True:</code>
2	<code> if button == False:</code>
3	<code> GPIO.output(24, True)</code>
4	<code> cam = cv2.VideoCapture(0)</code>
5	<code> ret, frame=cam.read()</code>
6	<code> cv2.imwrite('img2.jpg', frame)</code>
7	<code> img = cv2.imread("img2.jpg")</code>
8	<code> gray_img = cv2.cvtColor(img,</code> <code> cv2.COLOR_BGR2GRAY)</code>
9	<code> cv2.imshow("Hasil", img)</code>

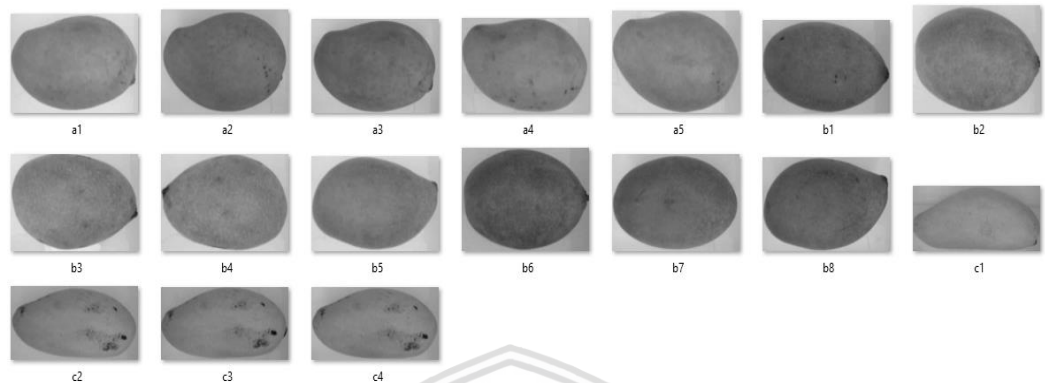
5.2.3.3 Implementasi Kode Program Pembuatan *Template*

Pada tahap implementasi kode program pengambilan *template* ini bertujuan untuk merealisasikan kode program pengambilan *template*. Untuk membuat *template* ini tergolong sederhana yaitu kita harus mengumpulkan beberapa gambar dari jenis yang akan kita gunakan untuk mendeteksi jenis buah mangga lalu mengubahnya kedalam citra *gray*. Pada tabel 5.5 dapat dijelaskan pada baris ke-1 program akan terus diulang – ulang dan akan berhenti apabila pengguna memberikan interrupt. Pada baris ke-2 jika *push button* ditekan maka akan menjalankan kode program dibawahnya. Pada baris ke-3 merupakan kode untuk menyalakan LED proses untuk memberi tahu bahwa proses sedang berjalan. Pada baris ke-4 merupakan kode program untuk mengakses kamera. Pada baris ke-5 merupakan pembacaan frame tiap detik dari kamera. Pada baris ke-6 mendeklarasikan variable *i* bernilai 0. Baris ke 7 merupakan kode program untuk mengambil 1 frame gambar dan menyimpan gambar tersebut dengan nama berurutan dari a0, a1, a2 sampai seterusnya. Baris ke-8 berfungsi untuk mengubah gambar yang telah diambil tersebut ke *gray*. Pada baris ke -10 merupakan increment nilai *i* sebanyak 1



Gambar 5.10 Hasil Beberapa Gambar *Template RGB*

Pada Gambar 5.10 merupakan gambar *template RGB* dari citra buah mangga. Selanjutnya akan diubah ke citra *gray* agar dapat diproses menggunakan *template matching*



Gambar 5.11 Hasil Beberapa Gambar *Template Grayscale*

Pada Gambar 5.11 merupakan gambar *template gray* dari citra buah mangga. Selanjutnya *template* sudah dapat diproses menggunakan *template matching*.

Tabel 5.5 Kode Pemrograman Pembuatan *Template*

Baris	Kode Program
1	while True:
2	if button == False:
3	GPIO.output(24, True)
4	cam = cv2.VideoCapture(0)
5	ret, frame=cam.read()
6	i=0
7	cv2.imwrite('a + str(i).jpg', frame)
8	img = cv2.imread("a + str(i).jpg")
9	gray_img = cv2.cvtColor(img,
	cv2.COLOR_BGR2GRAY)
10	i=i+1

5.2.3.4 Implementasi Kode Program *Template Matching*

Pada tahap implementasi kode program *template Matching* ini bertujuan untuk merealisasikan kode *template Matching*. Pada tabel 5.6 menjelaskan proses *template matching* pada citra untuk mendeteksi jenis buah mangga dengan menggunakan 1 gambar *template* untuk masing masing jenis. Pada baris 1-4. 6-9. 11-14 merupakan kode untuk membaca *template* masing masing jenis buah kemudian membaca gambar yang didapat dari *webcam*. Setelah itu setiap nilai piksel pada masing – masing gambar dimasukkan kedalam sebuah *array*. Kemudian *array* akan dicocokkan antara gambar dan *template* dengan menggunakan fungsi *cv2.TM_CCOEFF_NORMED* yang telah tersedia di dalam

library *Opencv*. Jika kecocokan diatas 78% maka akan menjalankan fungsi *for* dibawahnya yang akan memberi gambar berbentuk persegi dan nama jenis buah mangga pada gambar dan juga akan menyalakan *LED* sesuai dengan jenis buah mangga yang terdeteksi. Pada baris ke 37 mematikan led proses. Pada baris 38-40 menampilkan hasil pada layar selama 10 detik.



Gambar 5.12 Hasil Gambar Deteksi Jenis Mangga Manalagi

Gambar 5.12 merupakan citra buah mangga yang terdeteksi oleh sistem pendeteksi dengan jenis Mangga Manalagi



Gambar 5.13 Hasil Gambar Deteksi Jenis Mangga Arumanis

Gambar 5.13 merupakan citra buah mangga yang terdeteksi oleh sistem pendeteksi dengan jenis Mangga Arum Manis.



Gambar 5.14 Hasil Gambar Deteksi Jenis Mangga Golek

Gambar 5.14 merupakan citra buah mangga yang terdeteksi oleh sistem pendeteksi dengan jenis Mangga Golek

Tabel 5.6 Kode Pemrograman *Template Matching*

Baris	Kode Program
1	<code>template1 = cv2.imread("a1.jpg",</code>
2	<code>cv2.IMREAD_GRAYSCALE)</code>
3	<code>a, b = template1.shape[:::-1]</code>
4	<code>result = cv2.matchTemplate(gray_img, template1,</code>
5	<code>cv2.TM_CCOEFF_NORMED)</code>
6	<code>loc = np.where(result >= threshold)</code>
7	<code>templateb1 = cv2.imread("b1.jpg",</code>
8	<code>cv2.IMREAD_GRAYSCALE)</code>
9	<code>aa, bb = templateb1.shape[:::-1]</code>
10	<code>result = cv2.matchTemplate(gray_img, templateb1,</code>
11	<code>cv2.TM_CCOEFF_NORMED)</code>
12	<code>lak = np.where(result >= threshold)</code>
13	<code>templatec1 = cv2.imread("c1.jpg",</code>
14	<code>cv2.IMREAD_GRAYSCALE)</code>
15	<code>aaa, bbb = templatec1.shape[:::-1]</code>
16	<code>result = cv2.matchTemplate(gray_img, templatec1,</code>
17	<code>cv2.TM_CCOEFF_NORMED)</code>
18	<code>lik = np.where(result >= threshold)</code>
19	<code>for pt in zip(*loc[:::-1]):</code>
	<code>font = cv2.FONT_HERSHEY_SIMPLEX</code>
	<code>cv2.putText(img, 'Arum Manis', (50,50), font, 2,</code>
	<code>(0,0,255), 7, cv2.LINE_AA)</code>
	<code>GPIO.output(25, True)</code>


```
20 if GPIO.output(25, False):
21     for pt in zip(*lak[::-1]):
22         font = cv2.FONT_HERSHEY_SIMPLEX
23         cv2.putText(img, 'Manalagi', (50,50), font, 2,
24                     (0,0,255), 7, cv2.LINE_AA)
25         GPIO.output(22, True)
26 else:
27     for pt in zip(*loc[::-1]):
28         font = cv2.FONT_HERSHEY_SIMPLEX
29         cv2.putText(img, 'Arum Manis', (50,50), font,
30                     2, (0,0,255), 7, cv2.LINE_AA)
31         GPIO.output(25, True)
32     for pt in zip(*lik[::-1]):
33         font = cv2.FONT_HERSHEY_SIMPLEX
34         cv2.putText(img, 'Golek', (50,50), font, 2,
35                     (0,0,255), 7, cv2.LINE_AA)
36         GPIO.output(27, True)
37 GPIO.output(24, False)
38 cv2.imshow("Hasil", img)
39 cv2.waitKey(10000)
40 cv2.destroyAllWindows()
```



BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas proses pengujian serta menganalisis hasil dari pengujian yang dilakukan berdasarkan sistem yang telah dibuat. Adapun tujuan dilakukannya pengujian ada untuk mengetahui apakah semua kebutuhan yang diharapkan telah terpenuhi oleh sistem. Proses pengujian yang dilakukan yakni berupa pengujian fungsional, pengujian akurasi dan pengujian kecepatan pemrosesan sistem, dimana pengujian fungsional yakni menguji fungsi dari perangkat keras dalam hal ini berupa alat yang digunakan serta Kamera apakah dapat bekerja sesuai spesifikasinya, pengujian akurasi yakni menguji seberapa akurat sistem yang telah dirancang dan diimplementasikan dibandingkan nilai atau hasil yang sebenarnya, sedangkan pengujian kecepatan pemrosesan sistem yakni untuk menguji waktu pemrosesan ketika sistem mulai di jalankan hingga menghasilkan jenis klasifikasi. Berikut dijelaskan beberapa skenario pengujian yang dilakukan untuk menguji sistem.

6.1 Pengujian Kamera Logitech C270

Kamera adalah sensor utama dalam sistem ini yang berfungsi untuk mengambil citra dari buah mangga yang akan dideteksi. Pada pengujian kamera ini akan dilakukan dengan melakukan pengambilan citra dari beberapa jenis buah mangga yang berbeda dengan menggunakan kamera, kemudian mengukur tingkat keberhasilan kamera dalam mengambil gambar.

6.1.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui tingkat keberhasilan kamera dalam mengambil citra suatu objek. Dengan melakukan perbandingan keberhasilan kamera dalam mengambil citra dengan kegagalan kamera dalam mengambil gambar, maka akan diperoleh selisih nilai yang kemudian dapat diketahui nilai error dari kamera yang digunakan.

6.1.2 Prosedur Pengujian

Berikut prosedur yang dilakukan untuk menguji Raspberry Pi :

1. Meremote Raspberry Pi dengan laptop.
2. Membuka aplikasi Remote Desktop Connection.
3. Menghubungkan Kamera dengan Raspberry Pi melalui port USB
4. Menghubungkan push button dengan raspberry pi melalui pin GPIO
5. Menjalankan kode program pengambilan gambar dengan python yang sudah dibuat dengan Raspberry Pi.
6. Menekan push button pada sistem.
7. Mengamati hasil pengambilan gambar pada laptop dan mengukur tingkat keberhasilan kamera mengambil gambar dengan cara menghitung

perbandingan jumlah keberhasilan dan kegagalan kamera dari 10 percobaan dalam mengambil gambar.

Adapun cara untuk mengukur persentase keberhasilan yaitu dengan menggunakan **Persamaan (6.1)** berikut :

$$\text{Persentase Keberhasilan} = \frac{\text{jumlah berhasil}}{\text{jumlah percobaan}} \times 100\% \quad (6.1)$$



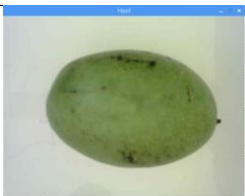
Adapun cara untuk mengukur persentase error yaitu dengan menggunakan **Persamaan (6.2)** berikut :

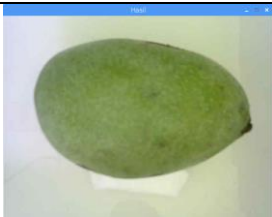
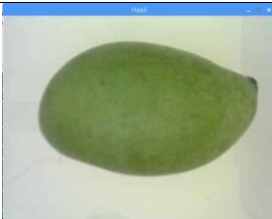
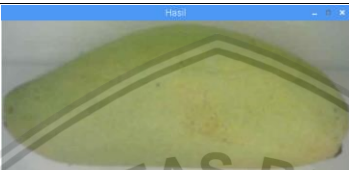
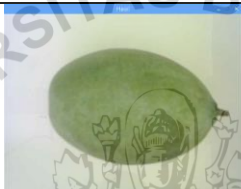


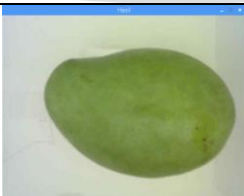
$$\text{Persentase error} = \frac{\text{jumlah gagal}}{\text{jumlah percobaan}} \times 100\% \quad (6.2)$$

6.1.3 Hasil dan Analisis Pengujian

Hasil pengujian kamera Logitech C270 ditunjukkan pada Tabel 6.1 di bawah, berdasarkan tabel tersebut terlihat bahwa kamera yang digunakan sukses mengambil 10 gambar dari beberapa jenis buah mangga yang berbeda dari 10 percobaan yang dilakukan.

Tabel 6.1 Hasil pengujian pengambilan gambar dengan kamera Logitech C270

Pengujian ke-	Hasil	Keterangan
1		Berhasil
2		Berhasil
3		Berhasil

4		Berhasil
5		Berhasil
6		Berhasil
7		Berhasil
8		Berhasil
9		Berhasil
10		Berhasil

Berdasarkan **Persamaan (6.1)** untuk menentukan persentase keberhasilan pada setiap pengujian, diperoleh nilai 100%. Contoh perhitungan persentase keberhasilan pada pengujian kamera sebagai berikut :

Jumlah percobaan= 10

Jumlah pengambilan gambar berhasil = 10

Jumlah pengambilan gambar gagal = 0

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah Berhasil}}{\text{Jumlah Percobaan}} \times 100\%$$

$$\text{Persentase Keberhasilan} = \frac{10}{10} \times 100\%$$

$$\text{Persentase Keberhasilan} = 100\%$$

Adapun untuk menghitung nilai rata-rata *error* keseluruhan pengujian adalah sebagai berikut:

$$\text{Persentase error} = \frac{\text{jumlah gagal}}{\text{jumlah percobaan}} \times 100\%$$

$$\text{Persentase error} = \frac{0}{10} \times 100\%$$

$$\text{Persentase error} = 0\%$$

Kamera Logitech C270 yang digunakan oleh *sistem* untuk mengambil citra mangga memiliki tingkat keberhasilan mencapai 100 persen. Dari 10 kali percobaan kamera dari sistem dapat mengambil gambar seluruhnya.

6.2 Pengujian LED

LED merupakan komponen yang berfungsi sebagai hasil output dari klasifikasi jenis buah mangga. Selain sebagai output, LED pada sistem juga digunakan sebagai lampu indikator penanda bahwa sistem sedang bekerja/memproses pengklasifikasian menggunakan *template matching*.

6.2.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui fungsionalitas dari LED yang digunakan apakah sesuai dengan fungsinya.

6.2.2 Prosedur Pengujian


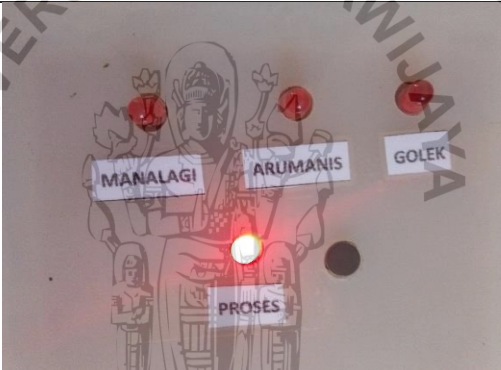

Berikut prosedur yang dilakukan untuk pengujian fungsionalitas dari LED :



1. Meremote Raspberry Pi dengan laptop.
2. Menghubungkan LED dengan pin GPIO pada Raspberry Pi
3. Menjalankan kode program dari LED.

4. Mengamati tampilan pada LED dengan memberikan kondisi yang berbeda-beda saat melakukan pengujian untuk melihat perbedaan tampilan dari LED ketika adanya kondisi yang berbeda

6.2.3 Hasil dan Analisis Pengujian

Tabel 6.2 Hasil pengujian LED

NO	Kondisi	Gambar	Keterangan
1	Ketika alat dalam keadaan <i>idle</i>		Kondisi awal sistem
2	Ketika ada terdapat penekanan triger oleh pengguna		Jika user menekan tombol trigger maka led akan menyala dan alat akan memproses.
3	Hasil Klasifikasi		Hasil Pendeteksian manga jenis Manalagi

4	Hasil Klasifikasi		Hasil Pendeteksian manga jenis Arumanis
5	Hasil Klasifikasi		Hasil Pendeteksian manga jenis Golek

Berdasarkan hasil pengujian tampilan LED sesuai yang ditunjukkan pada tabel 6.2 diatas terlihat bahwa LED dapat berjalan sesuai dengan yang diharapkan dengan kondisi-kondisi fungsi yang berbeda sekalipun.

6.3 Pengujian Akurasi Klasifikasi Template Matching

Sistem Pendeteksi Jenis buah mangga menggunakan metode *template matching* ini mempunyai tujuan utama untuk dapat mengklasifikasikan jenis – jenis buah mangga yang diletakkan kedalam sistem oleh pengguna. Oleh karena itu perlu diketahui tingkat keakuratan sistem dalam melakukan klasifikasi.

6.3.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui tingkat akurasi penggunaan metode *Template Matching* pada sistem pendeteksi jenis buah mangga yang telah dibuat.

6.3.2 Prosedur Pengujian







Prosedur pengujian akurasi *Template Matching* dilakukan dengan cara mengambil citra buah mangga yang akan diuji dengan cara meletakkan ke area tempat pengambilan gambar pada sistem. Sistem akan melakukan pengklasifikasian dengan metode *template matching* yaitu membandingkan gambar dengan *template* yang telah dibuat. Jumlah *template* sebanyak 39 gambar, dan data uji sebanyak 24 buah mangga dengan berbagai jenis. Untuk menentukan nilai ke akurasian dari sistem yaitu dengan cara membandingkan





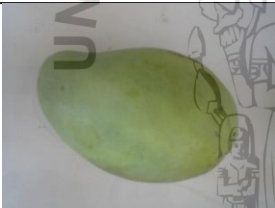



hasil klasifikasi yang dilakukan oleh sistem dengan hasil jenis buah mangga yang didapatkan dari pedagang buah mangga, rumus yang digunakan untuk menghitung akurasi ditunjukkan pada Persamaan (6.3).



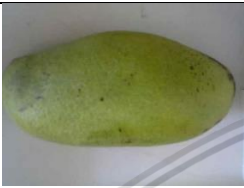





$$akurasi = \frac{total\ data - total\ data\ tidak\ sesuai}{total\ data} \times 100\% \quad (6.3)$$



6.3.3 Hasil dan Analisis Pengujian Klasifikasi *Template Matching*

Tabel 6.3 Hasil pengujian Klasifikasi *Template Matching*

Pengujian Ke-	Gambar	Kelas	Hasil Sistem	Kesesuaian
1		Arumanis	Arumanis	Sesuai
2		Arumanis	Arumanis	Sesuai
3		Manalagi	Arumanis	Tidak sesuai
4		Manalagi	-	Tidak terdeteksi
5		arumanis	Arumanis	Sesuai
6		golek	-	Tidak terdeteksi

7		Manalagi	manalagi	Sesuai
8		Manalagi	Manalagi	Sesuai
9		arumanis	Arumanis	Sesuai
10		Manalagi	manalagi	Sesuai
11		Arumanis	Arumanis	Sesuai
12		Arumanis	Arumanis	Sesuai
13		Manalagi	Manalagi	Sesuai
14		golek	golek	Sesuai

15		arumanis	manalagi	Tidak sesuai
16		Manalagi	Manalagi	Sesuai
17		golek	golek	Sesuai
18		Manalagi	arumanis	Tidak sesuai
19		Manalagi	-	Tidak terdeteksi
20		golek	-	Tidak terdeteksi
21		golek	golek	Sesuai
22		golek	golek	Sesuai

23		Manalagi	Manalagi	Sesuai
24		arumanis	Arumanis	Sesuai

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 6.3 terlihat bahwa dari jumlah 24 data terdapat 7 hasil dari sistem yang tidak sesuai dengan kelas sebenarnya. Sehingga akurasi yang diperoleh Sistem Pendeteksi Jenis Bua Mangga Menggunakan *Template Matching* ini berdasarkan Persamaan (6.3) adalah sebesar 70,83 %. Proses perhitungan akurasinya sebagai berikut :

$$akurasi = \frac{total\ data - total\ data\ tidak\ sesuai}{total\ data} \times 100\%$$

$$akurasi = \frac{24 - 7}{24} \times 100\%$$

$$akurasi = 70,83\%$$

Buah mangga dengan jenis manalagi memiliki keberhasilan 6 kali dari 10 kali percobaan, sehingga memiliki akurasi sebesar 60 persen. Kegagalan yang sering terjadi yaitu tidak terdeteksinya jenis buah mangga. Besar kemungkinan hal ini terjadi karena template yang disediakan terlalu sedikit.

Buah mangga dengan jenis arumanis memiliki keberhasilan 7 kali dari 8 kali percobaan, sehingga memiliki akurasi sebesar 87,5 persen. Kegagalan yang terjadi yaitu kesalahan hasil klasifikasi dengan keluaran buah mangga manalagi yang seharusnya arumanis. Besar kemungkinan hal ini terjadi karena kemiripan bentuk dari buah mangga arumanis yang dideteksi dengan buah mangga jenis manalagi.

Buah mangga dengan jenis golek memiliki keberhasilan 4 kali dari 6 kali percobaan, sehingga memiliki akurasi sebesar 66,6 persen. Kegagalan yang sering terjadi yaitu tidak terdeteksinya jenis buah mangga. Besar kemungkinan hal ini terjadi karena template yang disediakan terlalu sedikit dan adanya bintik – bintik kecil berwarna hitam yang terdapat pada kulit buah mangga.

6.4 Pengujian Waktu Komputasi Pemrosesan Sistem

6.4.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui waktu yang dibutuhkan sistem untuk melakukan proses pengklasifikasian jenis buah mangga menggunakan metode *Template Matching*, hal ini diperlukan untuk mengetahui performansi dari sistem yang telah dibuat.

6.4.2 Prosedur Pengujian

Untuk melakukan pengujian waktu komputasi pemrosesan sistem dilakukan dengan cara mengukur waktu komputasi ketika program dimulai sampai ketika program selesai dalam satu siklus sebanyak 24 kali pengujian. Pengujian waktu komputasi dengan menambahkan beberapa kode program. Prosedur pengujian ini dilakukan dengan menerapkan fungsi `time.time()` pada python. Program berupa penandaan waktu awal dan waktu akhir. Waktu awal yaitu pada saat dimana program berjalan ketika menerima trigger dari pengguna, dan waktu akhir yaitu ketika sistem berhasil mengeluarkan klasifikasi ke LED. Waktu komputasi didapat dengan cara waktu akhir dikurangi waktu awal. Implementasi fungsi `time.time()` pada python ditunjukkan pada Tabel 6.4 berikut.

Tabel 6.4 Kode Program Pengujian Waktu Komputasi

No	Kode Program
1	<code>import RPi.GPIO as GPIO</code>
2	<code>import time</code>
3	<code>import cv2</code>
4	<code>import os</code>
5	<code>import numpy as np</code>
6	
7	<code>==== //inisialisasi port====</code>
8	
9	<code>while True:</code>
10	<code> if button == False:</code>
11	<code> start=time.time()</code>
12	<code> print("Timer Started")</code>
13	
14	<code> GPIO.output(24, True)</code>
15	<code> cam = cv2.VideoCapture(0)</code>
16	<code> ret, frame=cam.read()</code>
17	
18	<code>==//Pengambilan Keputusan dengan Template Matching ==</code>
19	
20	<code>GPIO.output(24, False)</code>
21	<code>Finish=time.time()</code>
22	<code>print("Timer Stopped")</code>
23	<code>print("Time: " + str(finish-start))</code>
24	<code>cv2.imshow("Hasil", img)</code>
25	<code>cv2.waitKey(10000)</code>
26	<code>cv2.destroyAllWindows()</code>

6.4.3 Hasil dan Analisis Pengujian

Tabel 6.5 Hasil pengujian waktu komputasi

No	Pengujian Ke-	Jenis Buah Mangga	Waktu Komputasi (detik)
1	Pengujian ke-1	Arumanis	3,738
2	Pengujian ke-2	Arumanis	4,122
3	Pengujian ke-3	Manalagi	7,041
4	Pengujian ke-4	golek	3,022
5	Pengujian ke-5	Arumanis	4,285
6	Pengujian ke-6	golek	2,973
7	Pengujian ke-7	Manalagi	4,877
8	Pengujian ke-8	Manalagi	3,889
9	Pengujian ke-9	Arumanis	4,223
10	Pengujian ke-10	Manalagi	3,354
11	Pengujian ke-11	Arumanis	3,857
12	Pengujian ke-12	Arumanis	2,906
13	Pengujian ke-13	Manalagi	4.268
14	Pengujian ke-14	golek	4,005
15	Pengujian ke-15	Arumanis	4,339
16	Pengujian ke-16	Manalagi	3,556
17	Pengujian ke-17	golek	3,771
18	Pengujian ke-18	Manalagi	4,978
19	Pengujian ke-19	Manalagi	3,390
20	Pengujian ke-20	golek	3,259
21	Pengujian ke-21	golek	3,518
22	Pengujian ke-22	golek	3,476
23	Pengujian ke-23	Manalagi	5,273
24	Pengujian ke-24	Arumanis	5,688
Rata-rata			4,075

Tabel 6.6 Hasil pengujian waktu komputasi berdasarkan jenis

No	Jenis Buah Mangga	Rata – rata waktu
1	Manalagi	4,514
2	Arumanis	4,144
3	Golek	3,432

Berdasarkan tabel 6.5 dan 6.6 hasil pengujian yang dilakukan sebanyak 24 kali, waktu komputasi sistem untuk melakukan pengambilan keputusan jenis buah mangga dengan rata-rata waktu sebesar 4,073 detik. Buah mangga dengan jenis manalagi memiliki rata – rata waktu pemrosesan 4,514. Buah mangga dengan jenis Arumanis memiliki rata – rata waktu pemrosesan 4,144. Buah mangga dengan jenis Golek memiliki rata – rata waktu pemrosesan 3,432. Perbedaan waktu komputasi disebabkan banyaknya template dari tiap – tiap jenis yang disediakan. Tampilan salah satu pengujian waktu komputasi pemrosesan sistem melalui *Terminal* terlihat seperti pada Gambar 6.1.

```

pi@raspberrypi: ~
File Edit Tabs Help
tes.py:12: RuntimeWarning: This channel is already in use, continuing anyway. Use
se GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(27, GPIO.OUT)
Timer Started
rm: cannot remove 'img2.jpg': No such file or directory
tes.py:9: RuntimeWarning: This channel is already in use, continuing anyway. Use
e GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(24, GPIO.OUT) # set GPIO24 as an output (LED)
tes.py:10: RuntimeWarning: This channel is already in use, continuing anyway. U
se GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(25, GPIO.OUT)
tes.py:11: RuntimeWarning: This channel is already in use, continuing anyway. U
se GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(22, GPIO.OUT)
tes.py:12: RuntimeWarning: This channel is already in use, continuing anyway. U
se GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(27, GPIO.OUT)
Timer Started
rm: cannot remove 'img2.jpg': No such file or directory
Corrupt JPEG data: 1 extraneous bytes before marker 0xd2
Timer Stopped
Timer= 4.2858500480651855
** (Hasil:4574): WARNING **: Error retrieving accessibility bus address: org.fre

```

Gambar 6.1 Hasil Pengujian Waktu Komputasi Pada terminal

BAB 7 PENUTUP

Bab ini memuat penarikan kesimpulan berdasarkan tahap-tahap yang telah dikerjakan sebelumnya. Selain itu pada bab ini pula peneliti menyampaikan saran yang diharapkan dapat digunakan untuk pengembangan penelitian yang serupa dengan penelitian ini selanjutnya.

7.1 Kesimpulan

Sesuai dengan rumusan masalah yang diajukan diawal penelitian serta berdasarkan hasil analisis dari pengujian yang dilakukan maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Kamera diletakkan pada penyangga dan menghadap ke bawah pada area tempat buah mangga. Ketika Push button ditekan maka kamera akan langsung mengambil gambar buah mangga yang ada diarea yang telah disediakan. Kamera Logitech C270 dapat mengambil semua gambar yang diperintahkan dengan baik, dengan keberhasilan mencapai 100%.
2. Pada penelitian ini sistem dibuat dengan menggunakan metode *template matching* yaitu dengan membandingkan nilai dari tiap – tiap piksel secara berurutan pada gambar capture dengan setiap template masing – masing dari jenis buah mangga. Jika persentase kemiripan gambar dengan templatanya diatas tresholdnya yaitu 78% maka gambar hasil capture akan diklasifikasikan sesuai dengan jenis buah mangga dari template yang sesuai.
3. Akurasi yang diperoleh Sistem Pendeteksi Jenis Buah Mangga dengan Metode *Template Matching* yang diuji dengan jumlah data template sebanyak 36 data dan data uji sebanyak 24 data adalah senilai 70.83%.
4. Performansi Sistem Pendeteksi Jenis Buah Mangga dengan Metode *Template Matching* mempunyai nilai kecepatan waktu pemrosesan rata-rata sebesar 4,075 detik dari 24 kali pengujian. Buah mangga dengan jenis manalagi memiliki rata – rata waktu pemrosesan 4,514. Buah mangga dengan jenis Arumanis memiliki rata – rata waktu pemrosesan 4,144. Buah mangga dengan jenis Golek memiliki rata – rata waktu pemrosesan 3,432. Perbedaan waktu komputasi disebabkan banyaknya template dari tiap – tiap jenis yang disediakan.

7.2 Saran

Adapun beberapa saran yang dapat dijadikan acuan untuk pengembangan penelitian ini ataupun penelitian yang serupa kedepannya adalah sebagai berikut:

1. Menggunakan metode atau algoritma klasifikasi lain untuk membandingkan metode atau algoritma manakah yang mempunyai tingkat keakuratan lebih tinggi.

2. Menggunakan sensor yang lebih khusus untuk mendeteksi Jenis buah mangga, seperti warna dll, sehingga akan lebih meningkatkan keakuratan dalam pengklasifikasian.
3. Pengambilan gambar objek ditambahkan penutup atau pelindung agar cahaya dari luar tidak dapat masuk, sehingga tidak mengganggu akurasi sistem.
4. Menambahkan tingkat kematangan dalam pendeteksian oleh sistem.



DAFTAR PUSTAKA

- AAK. 1997. *Panduan Lengkap Dari Pembibitan Hingga Pasca Panen Buah Mangga*. Yogyakarta: Kanisius Media
- Ashari. Sumeru. 2017. *Mangga: Dulu. Kini dan Esok*. Malang: Universitas Brawijaya Press
- Anggriyono. Efraim. Iwan Setyawan². Ivanna K. Timotius. 2015. *Pemanfaatan Metode Template Matching untuk Face Tracking secara Real Time di Ruang Tertutup*. Universitas Kristen Satya Wacana
- Bahri. R. S.. 2012. *Perbandingan Algoritma Template Matching dan Feature Extraction Pada Optical Character Recognition*. Bandung: Jurnal Komputer dan Informatika.
- Chrisdiwanto. Thomas Oddy. 2018. *Perancangan Sistem Deteksi dan Pengenalan Rambu Peringatan Menggunakan Metode Template Matching*. Malang: Universitas Brawijaya.
- Ir. Pracaya. 2011. *Bertanam Mangga*. Jakarta: Penebar Swadaya Grup
- Pamungkas. Tito Tri. R. Rizal Isnanto. dan Ajub Ajulian Zahra. 2014. *Pengenalan Plat Nomor Kendaraan Menggunakan Metode Template Matching Dan Jarak Canberra*. Semarang: Universitas Diponegoro
- Pratiwi. Winda. 2015. *Panen Besar Mangga Dalam Pot: Membahas tentang bisnis buah mangga dalam pot*. Jakarta: Lembar langit Indonesia Group
- Raspberry Pi 3. Retrieved <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> . (diakses pada 10 Februari 2018).
- Rezki Trianto. dkk. *Klasifikasi Huruf Katakana Dengan Metode Template Matching Correlation*. Malang: Universitas Brawijaya.
- Subkhan. Moh. Khayat. Yuliana Melita Pranoto². 2012. *Pengenalan Image Wajah Dengan Menggunakan Metode Template Matching*. Surabaya: Sekolah Tinggi Teknik Surabaya
- Umam. Chairul. 2015. *Deteksi Osteoporosis Dengan Metode Template Matching Pada Citra Sinar Rontgen Tulang Panggul Manusia*. Semarang: Universitas Dian Nuswantoro

Wardhana. Adhitya Wishnu. Yudi Prayudi. 2008. *Penggunaan Metode Template Matching Untuk Identifikasi Kecacatan Pada PCB*. Yogyakarta: Universitas Islam Indonesia

Webcam Logitech C270. (2016). Retrieved <https://www.logitech.com/en-us/product/hd-webcam-c270> (diakses pada 25 Januari 2018).

Yamasari. Yuni. Mahfud Setyo Widargo. 2012. *Aplikasi Identifikasi Garis-Garis Telapak Tangan Berbasis Template Matching*. Surabaya: Universitas Negeri Surabaya

